AUGMENTING COMPLEX SURFACES WITH PROJECTOR-CAMERA SYSTEMS

BY

BRETT R. JONES

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:

Professor Roy Campbell

ABSTRACT

Display and interaction real-estate is fundamentally limited by the size of the screen in traditional display devices. Projection-based interfaces allow interaction with appropriated, everyday, passive surfaces. Instead of restricting interaction to a single rectangle, projection-based augmented reality makes interaction ubiquitous, taking the computer out of the box. Extending interaction to almost any surface enables new possibilities in a host of new domains, including entertainment, consumer electronics, advertising, medicine and many more.

Overlaying virtual elements onto complex physical surfaces under real-world conditions, presents a variety of challenges, many which are open research problems. A wealth of previous work has focused on projection onto constrained surfaces like planes, parabolic curves, etc. In this research, we explore projection onto arbitrarily complex physical surfaces. We describe a complete, practical approach embodied in a projector-camera system. Our system is constructed of low-cost, commodity hardware and demonstrates some of the exciting applications of Spatial Augmented Reality. We present a novel initial homography estimation for interactive projector-camera calibration. We introduce a unique hybrid projector pattern codification technique for improved correspondences. Finally, we present interactive surface particles, which are a surface independent content representation that enables reusable interaction with complex physical surfaces.

This thesis outlines the fundamental preliminaries of projector-camera systems, the current state of the art, along with our design decisions, implementation details and results. Covered topics include camera calibration, structured light 3D scanning and surface augmentation. This work demonstrates the technical and practical feasibility of appropriating complex surfaces as information displays; hopefully encouraging a future where a display device is no longer a discrete entity, but where our environment is our display and interface.

Dedicated to my mother.

She always told me that I could do anything I wanted. And silly me, I believed her.

ACKNOWLEDGMENTS

This research was made possible by the support and guidance of my advisor Professor Roy Campbell and Professor Guy Garnett. They gave me the freedom to explore, the guidance to not get lost and the encouragement to keep on going. Professor David Forsyth was invaluable in his guidance on the structured light scanner, always instantly understanding the issue and suggesting solutions. I would also like to thank Professor Brian Bailey for honing my academic writing skills.

This work is the product of a unique research partnership with fellow master's student Rajinder Sodhi. In such an emerging field, we have been able to learn the fundamentals and have begun to push the boundaries of what is possible by pooling our resources and intellect. Our shared time at the University of Illinois and Walt Disney Imagineering has enabled us to accomplish systems based research that would not be possible individually. I will always admire his never ending enthusiasm, dedication and patience.

I would never have entered into this field without Mark Mine, Tom LaDuke and Charita Charter at Walt Disney Imagineering who showed me that magic is possible (with the right technology and the right artistic talent).

Finally, this thesis would never have been written without my fiancée Mallory Casperson, whose words of encouragement and cups of coffee made this thesis a reality.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION [1]

Imagine if every surrounding object was a smart-surface, which could display information and receive input. No longer limited by interacting with flat, rectangular displays, every object that was once passive can now be interactive. Novel display and interaction technology has enabled projecting digital content onto complex physical surfaces, freeing content from the confines of a limited, flat monitor display. These projection-based interfaces are becoming increasingly feasible for use in ubiquitous display and interaction as projectors decrease in size, cost and power consumption. Everyday, passive objects can be transformed into interactive display surfaces through the use of novel projector-camera systems based on Spatial Augmented Reality [1]. Walls, desks, foam core, clay models, wooden blocks, sand pits, among other materials, can be turned into interactive displays, allowing for a new range of interaction possibilities.

Existing research has shown the potential for interaction with 2D content on planar surfaces such as tables and walls [2, 3, 4], through projector-camera systems. Extending this work to complex objects allows for a host of new interaction possibilities as any surface suitable for projection becomes an interactive entity.

Complex surfaces pose a variety of technical and practical challenges that inhibit research and development progress in this area. Exploring the interaction potential of complex surfaces requires time-intensive, complicated projector-camera systems. In order to overlay virtual content onto a physical surface, the projector-camera system must be accurately calibrated, a three dimensional scan of the surface must be acquired and content must be formulated in a way that adapts to the underlying display surface.

This thesis describes a complete, practical approach to augmenting complex surfaces with virtual content, through a projector-camera system entitled the Surface Interaction Engine. While camera calibration is a heavily researched topic, a complete, robust and practical description of projector-camera calibration is lacking from the community. We describe an implementation of projector-camera calibration,

[1] Portions of this chapter were reprinted with permission. ©2010 IEEE. Reprinted, with permission, from the 8th IEEE International Symposium on Mixed and Augmented Reality, 2010, Build Your World and Play In It: Interacting with Surface Particles on Complex Objects, Jones, B.R., Sodhi, R., Campbell, R., Garnett, G., Bailey, B.P.

from fundamentals to results, that was intended for real world environments of varying scales. Our interactive projector-camera calibration features a novel, initial homography estimation, that enables faster calibration and larger working volumes than similar techniques. We then describe a unique hybrid gray code and sinusoidal projector pattern codification technique, that enables higher precision projector-camera correspondences. The correspondences are then triangulated to create a dense three dimensional model of the surface. We present results on the performance of the structured light system along with scanning results at a variety of scene scales.

After the system is calibrated and a scan is acquired, a display surface can be augmented through overlaying virtual objects onto the physical surface. We present interactive surface particles, which are a novel view independent and surface independent content representation. Interactive surface particles allow end users to build, map and play with complex physical surface content. The surface particles are represented as two dimensional textured quads that are constrained to the physical surface and have associated interaction logic and physical behaviors. With surface particles, content can be programmed once and then reused on any surface suitable for projection, enabling a unique, tangible and immersive experience. More information about the interaction possibilities enabled by the projector-camera system are covered in Sodhi's master's thesis [5].

The main contributions of this thesis are:

1. A complete explanation of projector-camera calibration, structured light and surface augmentation from fundamentals to implementation and results.

2. A novel, initial homography estimation for interactive projector-camera calibration that enables faster calibration and larger working volumes.

3. A hybrid Gray code and sinusoidal projector pattern codification technique for increased projector-camera correspondence precision.

4. A view independent and surface independent representation of content using interactive surface particles.

The remainder of the thesis is outlined as follows:

Chapter 2: Related Work describes the state of the art in Spatial Augmented Reality, projector-camera calibration, and structured light.

Chapter 3: Projector-Camera Fundamentals introduces the basic mathematical models of a camera and projector, including the notation used through the rest of the thesis.

Chapter 4: Projector-Camera Calibration describes a complete, practical, interactive projector-camera calibration process.

Chapter 5: Structured Light explains using a calibrated projector-camera system for dense, three dimensional reconstruction.

Chapter 6: Surface Augmentation covers the fundamentals of surface augmentation and introduces interactive surface particles.

In order to increase the accessibility of this thesis, footnotes indicate relevant functions in the popular OpenCV computer vision library [1].

[1] OpenCV: http://opencv.willowgarage.com/wiki/

CHAPTER 2

RELATED WORK [1]

2.1 Spatial Augmented Reality

Shader Lamps first explored overlaying digital content onto physical surfaces [6], an area now entitled Spatial Augmented Reality. Raskar introduced the concept of projecting a 3D digital model calibrated onto a physical display surface, effectively animating the texture of any surface [6]. Then, in the Dynamic Shader Lamps [7] project, users could apply virtual paint to physical objects with a stylus. However, the 3D representation of the model was acquired through a tedious process with a tracked stylus, limiting the overall complexity of the surface.

Research has extensively explored new ways to embed digital content in our environments. For example, researchers have explored applications that augment our office environment where general computation can take place on surrounding surfaces [8, 9, 10]. Documents, pictures and other 2D media can be interacted with on flat surfaces, however arbitrarily complex surfaces are not addressed. The Office of the Future project forecasted a vision where any surface could be augmented with virtual content and developed a real-time, 3D scanning system to acquire complex surface geometry [11]. Similarly, the Smart Projector described by Bimber et al. combined a camera with structured light to acquire geometric information about a surface and its surrounding environment. The projector determines all parameters necessary to geometrically warp and color correct the input image such that the final output image appears correct when projected onto non-trivial surfaces [12].

Existing research has begun to explore projecting interactive content onto physical surfaces that are complex and automatically scanned. The Illuminating Clay and SandScape projects [13] provided a tangible user interface [14] for the domain of landscape architecture through dynamically scanning and projecting content. Users could physically mold a landscape surface while receiving dynamic information

[1] Portions of this chapter were reprinted with permission. ©2010 IEEE. Reprinted, with permission, from the 8th IEEE International Symposium on Mixed and Augmented Reality, 2010, Build Your World and Play In It: Interacting with Surface Particles on Complex Objects, Jones, B.R., Sodhi, R., Campbell, R., Garnett, G., Bailey, B.P.

about the surface through a series of landscape analysis visualizations. Johnson et al. created a multi-projector display on a sand table for military planning applications used for strategic planning and visualizing troop movement [15]. Users could place foam core buildings at pre-marked locations that represent their real-life counterparts and interactively annotate the surface using a tracked wand. Similarly, sand could be replaced with a human body for visualizing medical information. Yasumuro et al. created a system for projecting an ultrasound visualization calibrated onto a human torso [16].

2.2 Projector Camera Calibration

The projects described above exemplify the possibilities of using projector-camera technology to create immersive, interactive experiences that break the limitations of a flat display surface. Many of these applications require a calibration process to compute the 3D position and orientation (extrinsics) of a projector and camera relative to the display surface, as well as their geometric and optical properties (intrinsics), see Chapter 3. This process is essential in many projector-camera systems that are used for 3D reconstruction, virtual reality and tiled displays. The resulting geometry of a scene captured by structured light systems greatly depends on the quality of their calibration.

A large body of work has focused on efficient methods to calibrate a camera [17, 18, 19]. The standard calibration procedure is to capture a series of images of a known object (e.g. a cube) and compute the camera matrix that minimizes the reprojection error [20, 21]. One popular approach finds the image of the absolute conic to determine the intrinsics of the camera [22]. Arguably the most popular technique for camera calibration is the Zhang based method [23]. This method observes a set of known checkerboard patterns in a variety of unknown orientations and uses the extracted corner points of the pattern to compute the projection matrix. As we will describe in detail in Chapter 4, this method can be used to calibrate both the camera and the projector.

Researchers have explored tiling together multiple projectors to create seamless high resolution displays [24]. These high resolution displays require precise sub-pixel calibration of the projectors to accurately overlap content from multiple projectors on a single display surface. A number of approaches have been used for calibrating fixed projectors (e.g. [25]). Many of these approaches rely on using a camera to relay the projector's image location. For a planar surface, one can assume the standard pin-hole model and calibrate a projector using a planar homography from a single camera or a tree of camera homographies for a multiple projectors [26, 27]. The PixelFlex project [28] was a multi-projector display system that was optimized for planar surfaces. Similar to the Everywhere Display [8], the system allowed users to change

the display layout for different applications by controlling the projector's pan, tilt and focus. However, if the display surface is complex in shape, a more precise calibration method is needed. One approach used by Raskar et al. [29] integrated information about the projector from a set of calibrated cameras observing a display surface. To calibrate the projector, a series of structured light patterns are projected to establish 3D world point correspondences to 2D pixels in the projector's image space.

Many of the techniques described thus far focus on calibration for a fixed projector. A series of approaches have also been proposed to automate projector calibration and continuously update a portable projector's orientation in real-time. (e.g. [30]). The iLamps [31] project used a single mobile unit that contained a projector, a camera and a series of sensors. With these mobile units, users could project content that adapted to a variety of surfaces while reorienting the projector pose in real-time. Similarly, Johnson and Fuchs [10] presented a technique that is able to automatically recalibrate a projector on arbitrary display surfaces without having to interrupt the projected image. A single calibrated camera is used to observe the entire display and a feature matching algorithm is used between the projector and camera to re-estimate the pose of the projector continuously. Lee et al. [32, 33] removed the need for a camera entirely by embedding photo sensors within a known display surface. The system can display a series of structured light patterns that the photo sensors record to identify its 2D location in the projector's image plane. The known object in 3D world space and the corresponding 2D locations in the projector's image plane are used to calibrate the projector.

2.3 Structured Light

For many projection applications, in order to augment the texture of a surface, the system must capture a 3D representation of the scene. Traditional passive stereo vision algorithms use two or more cameras for range finding. Stereo vision emulates the way humans perceive depth by finding correspondences between different camera images to triangulate the 3D position of an object. With structured light [34], one of the cameras is replaced with a projector. Structured patterns from the projector illuminate a surface which a camera records to define pixel correspondences between a camera pixel and a projector pixel, effectively allowing the camera to see for the projector. Active range scanning methods that use structured lighting do not require scenes to be textured for unique correspondences, are computationally inexpensive, robust, offer low hardware costs and return a dense 3D representation of the scene.

A large benefit of active range scanning methods is that the correspondence problem between the stereo pair can be alleviated by using structured coding patterns. A number of structured light patterns have been

created with different variations in coding strategies to help optimize the accuracy and resolution of the captured surfaces. One such category of patterns is based on temporal coding where patterns are successively combined to form a codeword. Temporal coding patterns can be classified into binary patterns, n-ary codes, phase shifted patterns, or hybrid techniques using spatial and temporal strategies [35]. Introduced by Posdamer [36] in 1981, binary patterns are composed of black and white illumination, which are recorded as zeros and ones by a camera and are used to encode pixel locations in the projector. The Gray-code patterns [37], are a more robust variation of the binary patterns which ensures that each stripe boundary never occurs in the same location. Although using Gray codes eases the segmentation of the patterns, its discrete nature limits its accuracy. To increase accuracy, continuous methods such as phase shifted sinusoids [38, 39] can be used to encode pixels down to sub-pixel levels.

Another drawback of Gray codes is that they typically require projecting n sequences of patterns to encode 2^n pixels. To reduce the number of patterns, an n-ary pattern can be used to increase the number of intensity levels that encode the stripe patterns [40]. For example, a color based stripe pattern can be used where each symbol in the coding alphabet is associated with an RGB color [41]. One downside of the n-ary pattern is that in order to ensure robustness, one must take into account the spectral responses of the camera and projector as well as the surface albedo. To reduce the complexity inherent in modeling for the radiometric properties of the surface and ensure sub-pixel accuracy, the Surface Interaction Engine uses a hybrid Gray code and phase shifted pattern for structured light (see Section 5.1). For a more in depth survey of coding patterns, the reader is encouraged to review [35].

Although the triangulation step for either passive or active range finding methods requires users to calibrate the camera and the projector [42, 43, 44], it is still possible to capture scene geometry using uncalibrated methods. For example, Furukawa et al. [45] proposed a system that is based on uncalibrated stereo methods for passive stereo systems such as the PhotoTourism project [46]. The authors extended passive stereo methods to support a large number of correspondences that were acquired by using structured light patterns. The system captured a dense 3D representation and simultaneously calibrated the projector and the camera by minimizing a normalized error function. While these systems removed the need for users to calibrate the projector and camera, many of these methods added a layer of complexity that often resulted in greatly increased computational cost.

So far, the structured light methods discussed above all depend on the display objects to remain statically fixed. As we have discussed earlier, if a projector is moved, a range of self-calibrating approaches can be used to realign the projected content to the physical display surface. Similarly, if the projector

remains fixed while the object is moved, continuously projected structured light patterns can allow for real-time tracking of objects in the scene. Methods to track objects with structured light in real time while making the patterns imperceptible to a viewer are a long term research goal within the structured light community. Raskar et al. [11] proposed a method where each structured light pattern could be embedded in a projected image with a successive image displaying an inverse pattern. Cotting et al. [47] proposed a similar technique for embedding imperceptible binary patterns into projected images by using a standard DLP projector. Their approach utilized the micro-mirror modulation inherent in DLP technology to slightly adjust the intensity of each pixel in the image. The encoded images are captured by a high speed camera that is synchronized to the projector and exposes for a short period. Although the technique limits the overall dynamic range of the projector and relies on controlled lighting conditions, the approach allows for simultaneous display and acquisition of scene geometry.

Another popular approach has been to project images in the visible light spectrum while projecting the structured patterns in infrared which are captured by an infrared camera [48]. Lee et al. created a hybrid infrared-visible light projector that used an LED light source to emit both visible and non-visible IR images [33] that could be seen by embedded IR sensors. The LumiNetra project introduced a standalone IR projector that contained eight slide patterns where each pattern was associated with a single Gray code. Combined with IR LEDs, the projector could reach tracking framerates of 500 Hz and track objects affixed with IR light sensors. The downside of these IR projectors was that they were unable to project visible light images and were solely used for high speed tracking. While we have only covered a few techniques for doing real-time structured light, a large body of literature exists in the range finding community [49, 50, 51, 52], and we leave incorporating these techniques for future work.

CHAPTER 3

PROJECTOR CAMERA FUNDAMENTALS

In order to understand the use of projector-camera systems in Spatial Augmented Reality, we must first understand some fundamental mathematical models of cameras and lens optics. The description in this chapter is intended to be a primer to the fundamentals of projector-camera systems as well as an introduction to the notation used throughout the rest of the document. More detailed explanations of computer vision fundamentals can be found in secondary texts, such as [1, 53, 54].

3.1 A Projector as a Camera

It is important to note that a projector can be viewed as the dual of a camera [6]. A camera records light bouncing off of an object in three dimensional space and records a two dimensional image of the acquired light. A projector starts with a two dimensional image and emits colored light which is cast onto three dimensional objects (typically planar). The internal optics of a camera and a projector are very similar and we can model a camera and a projector equivalently. Therefore, for the rest of this chapter, the reader should think of a projector and a camera as interchangeable. Figure 3.1 depicts this relationship.

3.2 Lens Model

The basic pinhole camera model describes a system where light is passed through a single point onto an image plane, see Figure 3.2. Given a point $p = [X\ Y\ Z]'$ on a physical object, the point will be projected onto the image plane at point $\mu = [u\ v]'$, see Equation 3.1. This idea forms the basis of projective geometry. Generally a camera is modeled using three main components: intrinsic, extrinsic and distortion parameters.

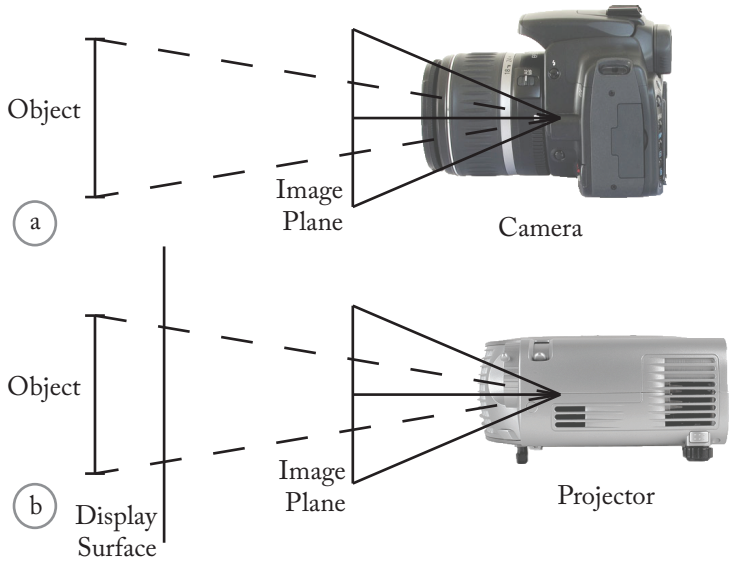

Figure 3.1: Projector-Camera Duals. (a) A camera observes a 3D object and records a 2D image. (b) A projector emits a 2D image onto a 3D display surface (typically planar).

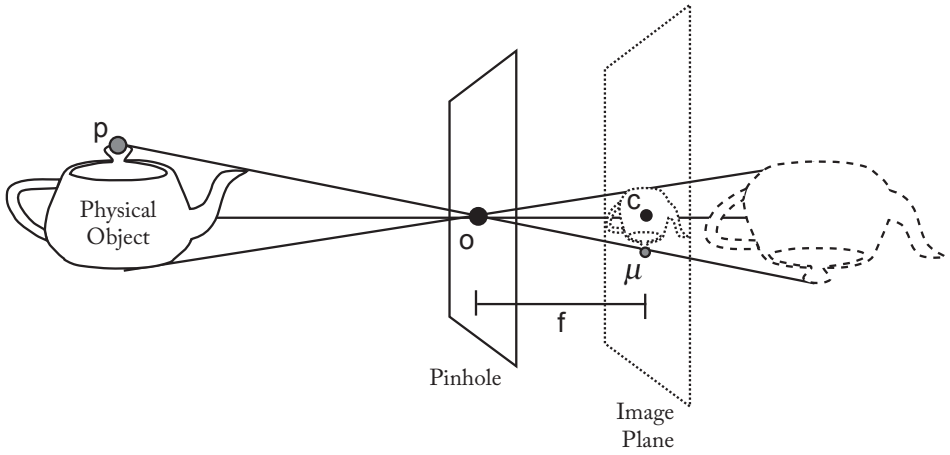

Figure 3.2: The pinhole imaging model. A physical object is imaged through a singe point (the pinhole, point o) onto an image plane. The physical point p is projected onto the image plane at point μ.

$$u = f\frac{X}{Z}$$
$$v = f\frac{Y}{Z} \tag{3.1}$$

3.2.1 Intrinsic Parameters

Intrinsic parameters define how the lens and sensor of the camera behave, excluding any non-linear distortion. The parameters change based on the camera and lens. The focal distance, f, defines the projection relation show in Figure 3.2. The principle point, c, defines the center of the image plane. As pixels can be rectangular as opposed to square, the model incorporates two scale parameters, a and b, that scale in the x and y pixel directions respectively. Finally, some camera models incorporate a skew parameter that allow for parallelogram pixels, θ, although for most modern cameras this term is negligible. These parameters are collected into a 3x3 matrix, K, that projects a point from camera centered world space into the image plane. It is important to note that this model assumes that the camera is in a canonical orientation, pointed down the positive Z axis.

$$K = \begin{bmatrix} af & -af\cot\theta & c_x \\ 0 & \frac{bf}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

As parameters f, a and b are not linearly independent we can substitute, $\alpha = af$ and $\beta = bf$, resulting in have five intrinsic parameters, $\{\alpha, \beta, \theta, c_x, c_y\}$.

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x \\ 0 & \frac{\beta}{\sin\theta} & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

To project a world space point, $p = [X, Y, Z]'$, through a canonically oriented camera to an image point, $\mu = [u, v]'$, we have,

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = \begin{pmatrix} K & \mathbf{0} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3.4}$$

3.2.2 Extrinsic Parameters

Unfortunately, our camera is not usually located in a canonical position and orientation. Therefore our model must incorporate the position and orientation of the camera. We introduce six additional parameters:

three defining the rotation, $\{\phi_1, \phi_2, \phi_3\}$ and three defining the translation, $\{t_x, t_y, t_z\}$. The angles of the rotation of the camera, ϕ_i, define a 3x3 rotation matrix, R, with rows r_i. These parameters are combined into a 3x4 matrix.

$$\begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} = \begin{bmatrix} \mathbf{r_1} & t_x \\ \mathbf{r_2} & t_y \\ \mathbf{r_3} & t_z \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_x \\ r_{2,1} & r_{2,2} & r_{2,3} & t_y \\ r_{3,1} & r_{3,2} & r_{3,3} & t_z \end{bmatrix} \tag{3.5}$$

To project a world space point, $p = [X, Y, Z]'$, through a non-canonically oriented camera to an image point, $\mu = [u, v]'$, we have:

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = K \begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.6}$$

Therefore we have eleven parameters,

$$\{\alpha, \beta, \theta, c_x, c_y, \phi_1, \phi_2, \phi_3, t_x, t_y, t_z\}. \tag{3.7}$$

3.2.3 Radial Distortion Parameters

Physical cameras and projectors frequently suffer from non-linear distortion that cannot be accounted for in our previous model. Radial distortion converts straight lines in a scene, to curved lines in the image, see Figure 3.3. These distortion effects can be quite severe, especially in less expensive web cameras.

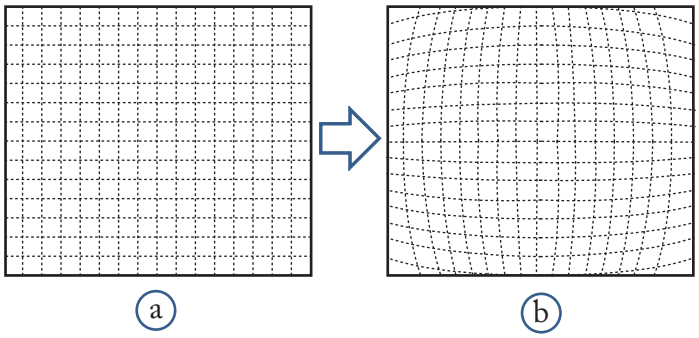

Figure 3.3: Radial distortion non-linearly warps an image turning (a) straight lines into (b) curved lines.

In order to compensate for this distortion we fit a polynomial distortion model with parameters $\{k_1, k_2, k_3, q_1, q_2\}$. We modify our previous model, projecting point p, to point $\mu = [u_2\ v_2]'$.

$$\begin{aligned} p' &= \begin{pmatrix} R & t \end{pmatrix} p \\ u &= \frac{p'_x}{p'_z} \\ v &= \frac{p'_y}{p'_z} \end{aligned} \tag{3.8}$$

$$\begin{aligned} u_1 &= u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2q_1 uv + q_2(r^2 + 2u^2) \\ v_1 &= v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + q_1(r^2 + 2v^2) + 2q_2 uv \\ &\text{where } r^2 = u^2 + v^2 \end{aligned} \tag{3.9}$$

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = K \begin{bmatrix} u_1 \\ v_2 \\ 1 \end{bmatrix} \tag{3.10}$$

3.3 Homography

If the scene is comprised of planar elements, a detailed model of intrinsics and extrinsics may be unnecessary. A simpler relation between two planes can be described by a single 3x3 matrix, H. Using a homography relation [26], any point on one plane $\mu_1 = [u_1,\ v_1]$ can be transformed into the equivalent point on the other plane $\mu_2 = [u_2,\ v_2]$.

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \tag{3.11}$$

These fundamentals do not represent a complete survey of camera modeling in computer vision. However, they should prepare the reader for the rest of the discussion and familiarize the reader with the notation that will be used.

CHAPTER 4

PROJECTOR CAMERA CALIBRATION

The most challenging portion of structured light scanning and surface augmentation involves determining the intrinsic, extrinsic and radial distortion parameters of the projector and the camera (see Section 3.2). Camera calibration typically involves observing a set of known correspondences or taking advantage of some known scene properties to determine the free parameters of the camera [22, 23]. In this chapter we introduce calibration fundamentals and describe a practical system for projector-camera calibration. We introduce a unique color pattern for color filtered Zhang based projector calibration of grayscale cameras. We also present a novel, initial homography estimation step that allows for an improved interactive calibration process.

4.1 Calibration Fundamentals

Camera calibration involves determining the unknown parameters, given some known information about the scene. As mentioned in Section 2.2, camera calibration is a very well studied topic and there are a wide variety of approaches for effectively determining the intrinsics and extrinsics of a camera [17, 42, 43, 23]. Assuming no radial distortion we have eleven unknown parameters, see Section 3.2,

$$\{\alpha,\ \beta,\ \theta,\ c_x,\ c_y,\ \phi_1,\ \phi_2,\ \phi_3,\ t_x,\ t_y,\ t_z\}. \tag{4.1}$$

The easiest way to calibrate a camera is to observe at least six known points in three dimensional world space coordinates, $[X_i\ Y_i\ Z_i]$, called a 3D rig calibration, see Figure 4.1. Observing six points in a two dimensional camera image, $[u_i\ v_i]$, yields twelve known variables. Solving a least squares solution, given at least six known point correspondences, yields our unknown camera parameters [55].

In order to solve for the camera parameters, a single projection matrix, A, is created that incorporates all of the parameters. This solution can then be constrained and the individual parameters can be extracted.

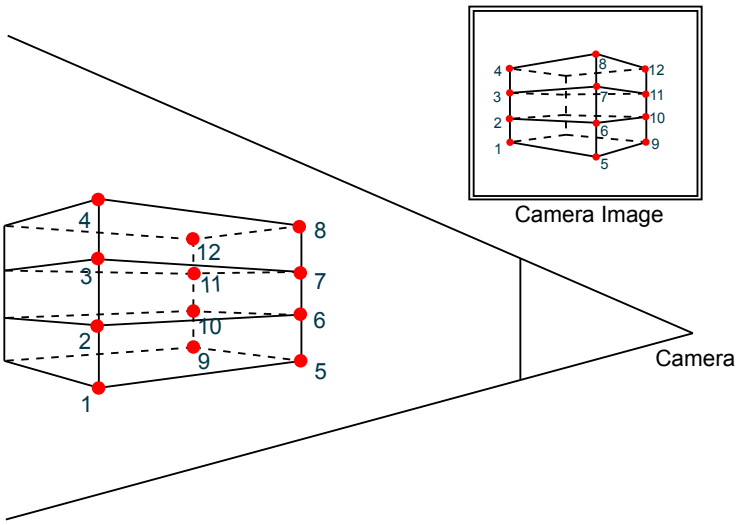

Figure 4.1: A 3D rig calibration. A set of known 3D world points are observed by a camera. The correspondences are used to solve for the intrinsic, extrinsic and distortion parameters.

$$K\begin{pmatrix} R & t \end{pmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & 1 \end{bmatrix} = A \tag{4.2}$$

The known correspondences result in a series of equations, which are defined up to a scale, w.

$$\begin{bmatrix} wu_i \\ wv_i \\ w \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{4.3}$$

$$\begin{aligned} wu_i &= a_{1,1}X_i + a_{1,2}Y_i + a_{1,3}Z_i + a_{1,4} \\ wv_i &= a_{2,1}X_i + a_{2,2}Y_i + a_{2,3}Z_i + a_{2,4} \\ w &= a_{3,1}X_i + a_{3,2}Y_i + a_{3,3}Z_i + 1 \end{aligned} \tag{4.4}$$

After substituting for w, we have the following,

$$\begin{aligned} 0 &= a_{1,1}X_i + a_{1,2}Y_i + a_{1,3}Z_i + a_{1,4} - u_i(a_{3,1}X_i + a_{3,2}Y_i + a_{3,3}Z_i + 1) \\ 0 &= a_{2,1}X_i + a_{2,2}Y_i + a_{2,3}Z_i + a_{2,4} - v_i(a_{3,1}X_i + a_{3,2}Y_i + a_{3,3}Z_i + 1) \end{aligned},$$

then we rewrite the equation in terms of $[u_i\ v_i]$,

$$u_i = a_{1,1}X_i + a_{1,2}Y_i + a_{1,3}Z_i + a_{1,4} - a_{3,1}(u_iX_i) - a_{3,2}(u_iY_i) - a_{3,3}(u_iZ_i)$$
$$v_i = a_{2,1}X_i + a_{2,2}Y_i + a_{2,3}Z_i + a_{2,4} - a_{3,1}(v_iX_i) - a_{3,2}(v_iY_i) - a_{3,3}(v_iZ_i).$$

and finally grouping by $a_{i,j}$ allows writing the equation as $PV = F$, where A are the unknown parameters,

$$V = [a_{1,1}\ a_{1,2}\ a_{1,3}\ a_{1,4}\ a_{2,1}\ a_{2,2}\ a_{2,3}\ a_{2,4}\ a_{3,1}\ a_{3,2}\ a_{3,3}]'. \tag{4.5}$$

The system of $2n$ equations can then be represented in matrix form.

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -u_2X_2 & -u_2Y_2 & -u_2Z_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -v_2X_2 & -v_2Y_2 & -v_2Z_2 \\ \vdots & & & & & \vdots & & & & & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix} \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{1,4} \\ \vdots \\ a_{3,2} \\ a_{3,3} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{bmatrix} \tag{4.6}$$

This system of equations can then be solved using linear least squares or other optimization methods. Finally, if the intrinsic and extrinsic must be known explicitly, the unknown eleven parameters $\{\alpha, \beta, \theta, c_x, c_y, \phi_1, \phi_2, \phi_3, t_x, t_y, t_z\}$, can be extracted from these eleven solved variables using known constraints. See [53] for a full treatment of camera calibration and for an explanation of solving for the radial distortion parameters in the model.

There are serious practical hindrances to 3D rig calibration. As we are fitting a model to some observed data in camera calibration, the error is smallest within the calibrated volume and expands rapidly outside of the calibrated volume. Therefore, if we are scanning a one meter tall object, we must create a calibration rig consisting of a set of known 3D world points on the scale of one cubic meter. Clearly, this method does not scale well and is a problem for practical in field use of projector-camera calibration. Additionally, creating an accurate 3D rig is a non-trivial construction task and construction errors will greatly affect the calibration process.

4.2 Zhang Camera Calibration

A widely established technique for practical camera calibration involves observing a planar checkerboard calibration target from a variety of views, often referred to as Zhang based calibration [1] [23]. The user must wave the planar checkerboard target, see Figure 4.2, in front of the camera obtaining images of the checkerboard pattern at differing positions and orientations, see Figure 4.3. We can use the corners of the checkerboard as known feature points which we can find in the acquired calibration images using a Harris detector[2] [56]. As printers are a commonplace item, users can print their own precise checkerboard patterns at low cost, enabling accurate, readily available calibration patterns.

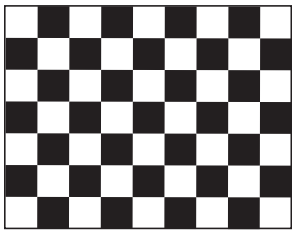

Figure 4.2: Example 9x7 calibration checkerboard pattern. Interest points (8x6) are the corners of the checkerboard pattern.

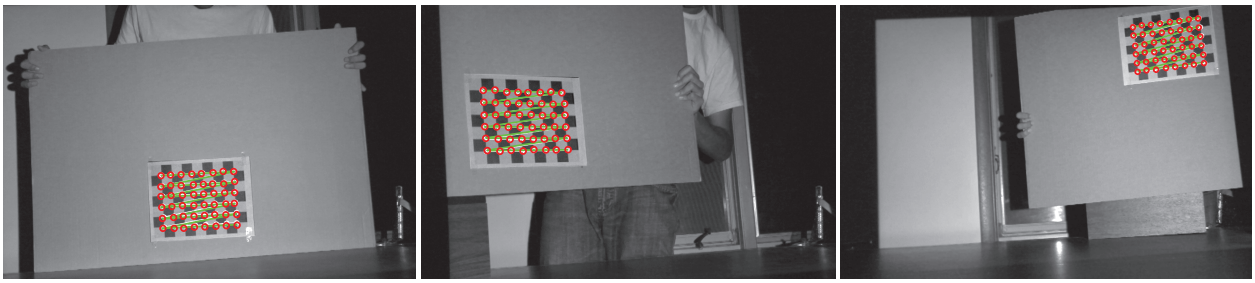

Figure 4.3: Zhang based camera calibration. A camera observes a checkerboard at varying orientations and the corners are tracked (shown in red).

The calibration process outlined in Section 4.1 involved correspondences between known 3D world points and tracked 2D correspondences. The Zhang method utilizes the constraint that the known 2D world points (the checkerboard corners) must all lie in a plane. Furthermore, as the checkerboard is observed at multiple orientations, the camera intrinsics must remain constant across the views. The intrinsic, extrinsic and radial distortion parameters are solved using a non-linear optimization procedure.

Due to the failure rate of the checkerboard tracking, not every picture of a checkerboard results in usable

[1] OpenCV reference: cvCalibrateCamera2
[2] OpenCV reference: cvCornerHarris, cvFindCornerSubPix

calibration data. In order to overcome this obstacle, the calibration process is made interactive, allowing the user to adjust the checkerboard if the tracking fails. The user views a video feed from the camera and when the system correctly detects all of the checkerboard feature points, a virtual marker is overlaid onto each positive corner detection, as shown in Figure 4.3. After a series of valid images are recorded (usually 5-15 images), camera calibration is run on the acquired data set.

4.3 Projector Calibration

Projector correspondences are much more difficult to acquire, as a projector cannot directly observe a scene. However, we can use a camera to observe the scene for the projector, see Figure 4.4. One approach starts with a camera observing a physical checkerboard pattern resulting in a set of world point to camera pixel correspondences, $X_w \Rightarrow x_c$. Then the methods described in Section 5.1 can be used to establish a dense correspondence mapping from camera pixels to projector pixels, $x_c \Rightarrow x_p$. Once these camera to projector correspondences are determined, checkerboard to camera correspondences can be transformed into checkerboard to projector correspondences, $X_w \Rightarrow x_p$. Then the projector can be calibrated using the Zhang based calibration method that was used to calibrate the camera. This approach was taken in [57].

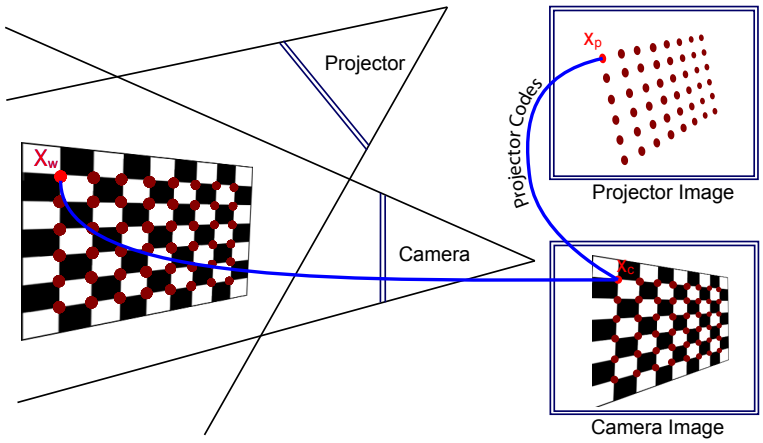

Figure 4.4: A physical checkerboard pattern with known size is observed by a camera and projector. First camera correspondences $X_w \Rightarrow x_c$ are established by finding the corners of the checkerboard (shown in red). Then camera to projector correspondences are established using techniques described in Section 5.1, yielding $x_c \Rightarrow x_p$ and thus $X_w \Rightarrow x_p$. These correspondences are then used to calibrate the projector.

Unfortunately, in order to create an accurate correspondence between a camera and a projector pixel, we typically have to temporally encode a sequence in patterned projection. If a user is holding the checkerboard, the pattern may move between frames, invalidating the code. Furthermore, these

correspondences are dependent on the position of the checkerboard and thus they must be determined at each target position/orientation. Requiring the user to hold the calibration pattern still and wait for these correspondences may be an unrealistic expectation for practical use.

Therefore the goal is to establish exact projector correspondences in a single pattern, to reduce artifacts from calibration target movement. In order to accomplish this, the calibrated camera can assist in the calibration of the projector. If a calibrated camera observes a checkerboard of known size on a plane, the resulting correspondences can be used to solve for the equation of the plane in world space, see Figure 4.5. Now given any point in the camera that falls onto this plane, the 3D world position of the point can be determined through ray-plane intersection. Therefore, if the projector emits a spatial code that the camera can locate, this point can be translated into the corresponding 3D world point. For instance, if the projector emits a checkerboard pattern, the corners of the pattern can serve as spatial codes yielding projector to world space correspondences. The projector can then be calibrated as previously mentioned.

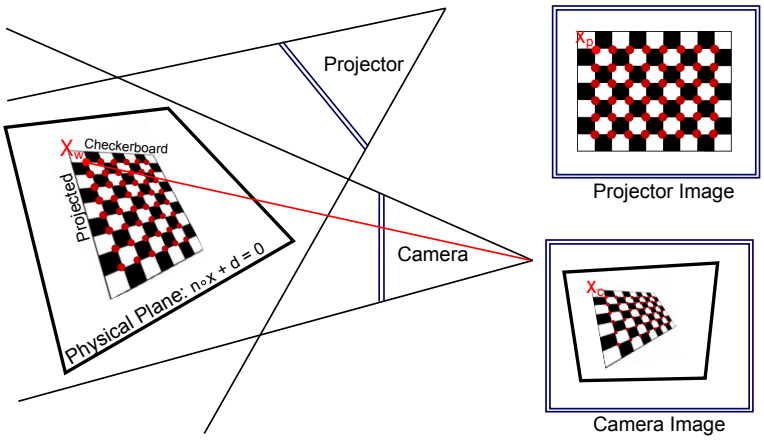

Figure 4.5: A virtual checkerboard is projected onto a known plane, $n \circ x + d = 0$. The camera observes this virtual checkerboard resulting in correspondences between projector pixels x_p and camera pixels x_c. Assuming a calibrated camera, the world points corresponding to the virtual checkerboard corners X_w can solved for with ray-plane intersection. This gives $X_w \Rightarrow x_p$ correspondences, which are used to calibrate the projector.

If the camera is using a physical checkerboard to locate the reference plane and the projector is emitting a virtual checkerboard, there is a potential for interference between the two signals [44]. For instance, a physical black and white checkerboard like the one in Figure 4.2, with a similar virtual checkerboard overlaid by projection would not be distinguishable by the camera. The signals can be separated by taking advantage of the three independent color channels (RGB) of the projector. For instance, if a red and black checkerboard is observed in red light, the camera will see a red and black checkerboard. If the red channel

of the captured image is viewed independently, there will appear to be a white and black checkerboard. However, if a red and black checkerboard is observed in green light, the camera will only see black. This approach is taken in [38, 43].

This technique can be extended to work with black and white cameras. We present a unique color filtered projector calibration checkerboard that appears to the grayscale camera as almost uniformly gray under white light, while appearing black and white under red illumination. The physical checkerboard is constructed of two colors that have roughly equal grayscale intensity, such as yellow and blue, see Figure 4.6. Under white illumination by the projector, both colors appear gray, however under red illumination, the checkerboard appears almost black and white.

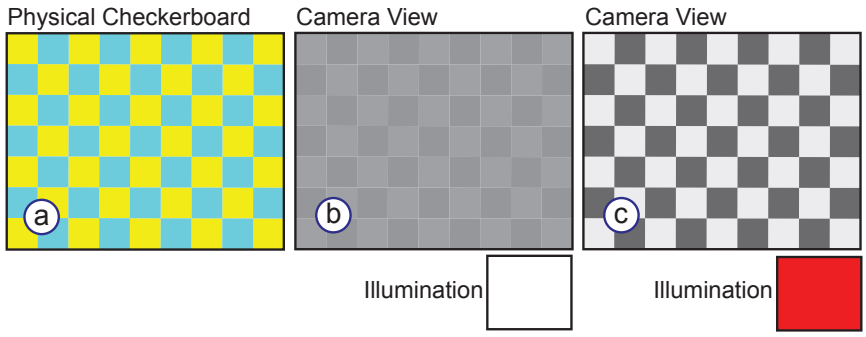

Figure 4.6: (a) An example yellow (RGB= [0.965, 0.922, 0.075]) and blue (RGB=[0.435, 0.800, 0.866]) checkerboard pattern. (b) The pattern appears almost uniform gray when observed by a grayscale camera under white illumination. (c) However the same pattern appears almost black and white under red illumination.

If a virtual black and white checkerboard is projected onto this yellow and blue physical checkerboard, the resulting camera image is a gray and black checkerboard corresponding to the virtual checkerboard. These results can be further improved by first capturing a camera image of the physical checkerboard under solid white illumination. Assuming a static physical checkerboard, the images under solid white illumination and under checkerboard illumination can be subtracted, yielding only the virtual checkerboard. This process is illustrated in Figure 4.7 with actual calibration images.

4.4 Simplified Interaction through Homography Estimation

The previous approach relies on a virtual checkerboard being projected onto the plane defined by a physical checkerboard. As with camera calibration, the projected checkerboard should be moved throughout the entire calibration volume. Unless the physical plane occupies the entire projected volume,

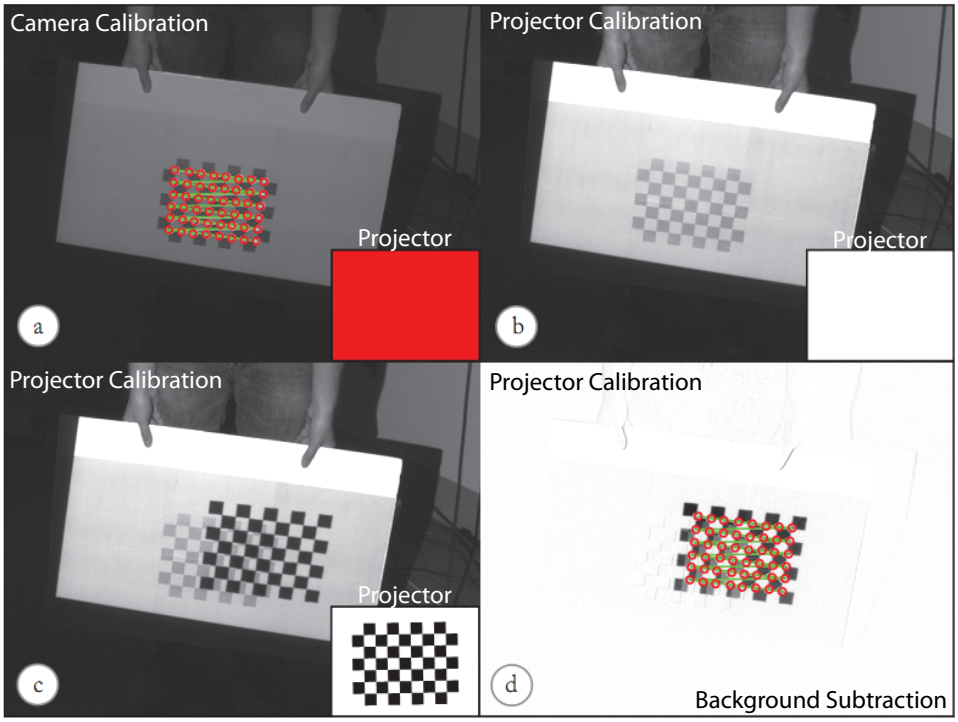

Figure 4.7: Practical projector-camera calibration using a yellow and blue physical checkerboard pattern. (a) The checkerboard is observed by a grayscale camera under red illumination and the resulting correspondences are used to calibrate the camera. (b) The same checkerboard is viewed under white illumination. (c) The projector emits a black and white virtual checkerboard pattern onto the plane with the physical yellow and blue checkerboard pattern. (d) Images (b) and (c) are subtracted resulting in only the virtual checkerboard pattern, which is used to calibrate the projector.

there will be sections of the calibration volume that are not covered by the physical plane and therefore there are areas of the calibration volume that cannot receive the projected pattern. Ideally the virtual checkerboard would be projected onto the plane in the same region as the physical checkerboard. Unfortunately, in order to project the virtual checkerboard onto the physical checkerboard, the projector and camera must be calibrated, which is precisely what we are attempting to accomplish.

We introduce a novel method for improved interactive projector-camera calibration. We created this method to decrease calibration time and increase the calibration volume. A related method, [58], utilized projected fiducial markers to estimate a homography. In our method, an approximation of where to project the virtual checkerboard is established through an initial homography between the camera and projector. This homography is created from a plane in the middle of the desired calibration volume. The system projects a virtual checkerboard onto the physical plane and the camera find the corners of the checkerboard pattern, see Figure 4.8. The homography, H_0, is then found by a least squares fit of the projector and camera correspondences.

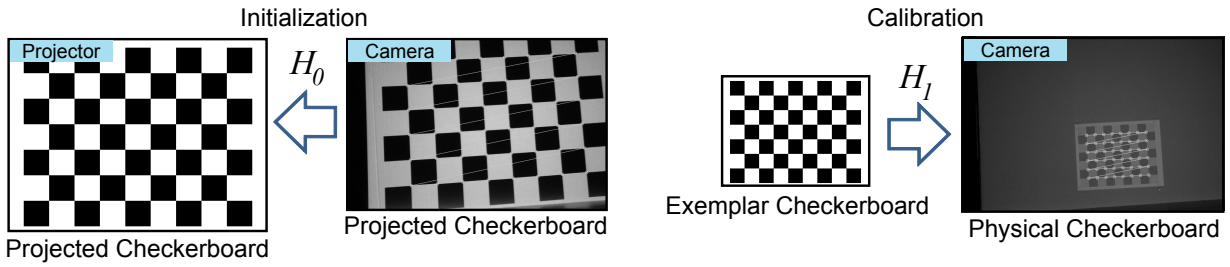

Figure 4.8: In order to roughly align the projected checkerboard with the physical checkerboard an initial homography is computed between the camera and projector. (left) A physical plane is place in the middle of the calibration volume. A projected checkerboard is then imaged onto the plane and observed by the camera yielding homography H_0. (right) A homography is established between an exemplar checkerboard and the image of the physical checkerboard, H_1. This allows for the creation of the idealized physical checkerboard, as seen from the camera.

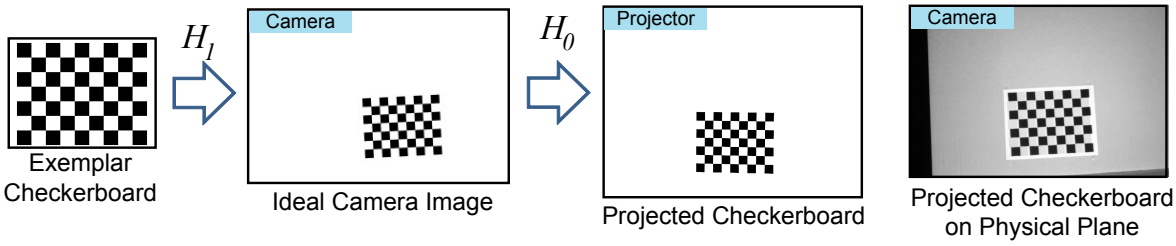

Figure 4.9: The projected virtual checkerboard is created using the two homographies H_0 and H_1 explained in Figure 4.8. First the exemplar checkerboard is warped using H_1 to create an idealized camera image. Then this image is warped using H_0 to create the projected checkerboard. The image on the right demonstrates the projected and physical checkerboards are aligned. Note the alignment is only close when the plane is near the initial plane position and orientation.

During projector-camera calibration, the homography is used to estimate the location and orientation of the physical plane. The camera first observes the physical checkerboard on the plane, locating the checkerboard corners. While these interest points could be used directly to generate a projected checkerboard, due to sampling and radial distortion errors the interest points no longer form a perfect checkerboard. Therefore another homography, H_1, between an exemplar checkerboard and the detected checkerboard corners is established, see Figure 4.8. Then an idealized camera checkerboard is generated by transforming the exemplar checkerboard with H_1. Then this checkerboard is transformed by H_0 yielding a projector checkerboard, see Figure 4.9. In practice this approximation greatly expedites the calibration process. As long as the physical plane containing the checkerboard is slightly larger than the actual checkerboard the approximation almost always places the projected checkerboard on the physical plane.

4.5 Results

As it is not possible to directly measure the intrinsic and extrinsic parameters of the projector and camera, the majority of previous work evaluate the calibration accuracy through evaluating the accuracy of a structured light scan using the calibration parameters (e.g. [23, 38, 43, 44]). Therefore, an example real world calibration data set is provided in Section 4.5.1 and Section 5.4 evaluates the accuracy of the overall system through a structured light scan.

4.5.1 Real World Data

The following results were captured using a Point Grey Firefly MC camera with a Tamron 13VM2811ASIR variable focus $(2.8 - 11\text{mm})$ lens. The camera resolution is $[640,\ 480]$ at 61 Hz. The projector was a Sharp XG-P10XU, with a $[1024,\ 768]$ resolution at 3000 lumens and a focal length of $(49.1 - 63.8\text{mm})$. The projector was located approximately 1.7 m away from the final calibration target and the camera was 1.5 m away from the final calibration target, with a distance of 0.8 m between the projector and camera. Note that the measurements are approximate as the extrinsic parameters measure the position and orientation of the camera center, which is physically unknown.

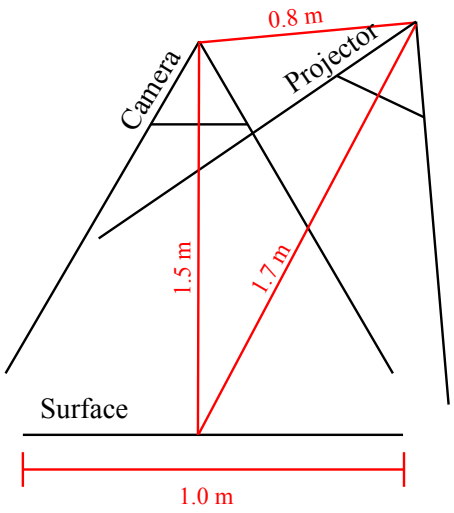

Figure 4.10: A real world example data set was acquired with a projector and camera pointed at a calibration volume of approximately 1 m^3.

The calibration set was comprised of 15 camera calibration images and 15 projector calibration images. Sample images are provided in Figure 4.7 (a) and (d). The calibration results are as follows, with ^{c} indicating camera parameters and ^{p} indicating projector parameters. For instance K^c is the 3x3 camera

intrinsic parameter matrix.

$$K^c = \begin{bmatrix} 1212.75 & 0.00 & 375.42 \\ 0.00 & 1213.67 & 219.34 \\ 0.00 & 0.00 & 1.00 \end{bmatrix} \qquad K^p = \begin{bmatrix} 2191.60 & 0.00 & 451.47 \\ 0.00 & 2173.62 & 453.68 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$r^c = \begin{bmatrix} 2.257 \\ 1.972 \\ 0.125 \end{bmatrix} \qquad r^p = \begin{bmatrix} -2.024 \\ -2.027 \\ 0.427 \end{bmatrix}$$

$$t^c = \begin{bmatrix} -209.766 \\ 142.366 \\ 1521.271 \end{bmatrix} \qquad t^p = \begin{bmatrix} -175.374 \\ -6.346 \\ 1706.441 \end{bmatrix}$$

$$k_1^c = -0.278 \qquad k_1^p = 0.362$$

$$k_2^c = -0.655 \qquad k_2^p = -7.272$$

$$k_3^c = 4.200 \qquad k_3^p = 44.827$$

$$q_1^c = -6.158 * 10^{-4} \qquad q_1^p = -0.027$$

$$q_2^c = -2.148 * 10^{-3} \qquad q_2^p = 3.327 * 10^{-3}$$

Projector-camera calibration is often regarded as the most difficult part of building any practical projector-camera system. Typically, the accuracy of the calibration greatly influences the performance of the overall application. Section 5.4 provides more information on the accuracy of the system and Section 6.5 demonstrates how the current accuracy is sufficient for user interaction. Now that we have a calibrated projector-camera system, we can turn our attention to dense three dimensional reconstruction and surface augmentation.

CHAPTER 5

STRUCTURED LIGHT

In order to overlay virtual objects onto a physical surface we must have an accurate 3D representation of the display surface. While there are many methods for acquiring the 3D geometry of the surface, structured light takes advantage of the existing projector-camera system and yields a relatively low noise, high resolution scan of the surface. In traditional stereo vision, two cameras observe feature points in the scene and then triangulate the 3D location of the shared feature points. In structured light, one of the cameras is replaced by a projector and dense feature correspondences are created through projection. The camera and projector are calibrated prior to the scanning, with known intrinsic, extrinsic and radial distortion parameters. Then the dense feature correspondences can be used to triangulate the 3D location of the feature points, see Figure 5.1. Unfortunately, this process is complicated by a lossy code transmission process and the discrete and inherent non-linear nature of projection and imaging devices.

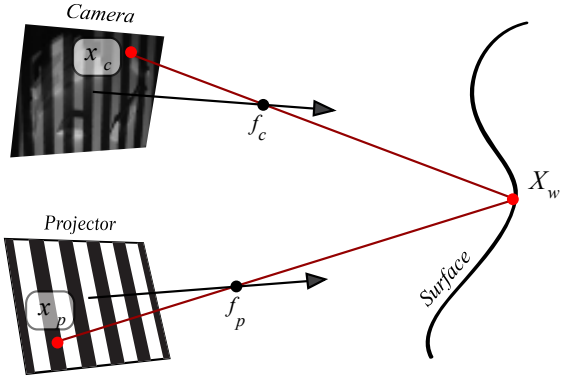

Figure 5.1: Structured light determines a 3D world point position X_w by triangulating the rays defined by a projector pixel x_p and its corresponding camera pixel x_c, whose correspondence is established through patterned projection.

Our projector-camera system, the Surface Interaction Engine, has an integrated structured light scanner for acquiring the 3D geometry. We describe the fundamentals of structured light along with the particular design choices of the authors. We present a unique hybrid Gray code and sinusoidal pattern codification process. Additionally, results from a wide variety of scene sizes are presented and the error in the system is

addressed. The implementation was initially based off code that was available as part of a SIGGRAPH 2009 course by Lanman and Taubin [59], although it has since been heavily modified. For a more rigorous explanation, see Batlle et al., an excellent survey in structured light [55].

5.1 Projector Pattern Codification

Many of the techniques mentioned rely on a dense set of correspondences between camera and projector pixels. In order to acquire these correspondences the projector can emit uniquely identifiable spatial or temporal codes which can be decoded by the camera. The goal is to uniquely identify a mapping from every camera pixel to a projector pixel. Typically this is accomplished by using a horizontal pattern to encode the projector row and a vertical pattern to encode the projector column. There are a wide variety of techniques for encoding information in projected patterns [60]. This thesis covers the two most widely used techniques, Gray codes and sinusoidal patterns, along with a unique fusion of these techniques.

5.1.1 Gray Codes

One code for establishing correspondences is a temporally multiplexed Gray code sequence. A Gray code sequence is similar to a binary sequence, however only one bit changes between successive numbers in the sequence.

$$\begin{array}{cccccccccc} \text{binary} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \text{gray} & 000 & 001 & 011 & 010 & 110 & 111 & 101 & 100 \end{array} \tag{5.1}$$

In order to encode the column indices, a temporal vertical Gray code pattern is projected. In this pattern, black corresponds to a 0 bit and white corresponds to a 1 bit. For instance, in Figure 5.2 the red square corresponds to col $= 14$ and receives the binary sequence 1101 and the gray sequence 1011. For a $[1024, 768]$ resolution projector image, the pattern must encode 10 bits for the row and 10 bits for the column, resulting in a total of 20 images.

When the camera receives this projected pattern it must then classify the results into a 0 or 1 bit for each camera pixel. The easiest method would start with a reference image of uniform white illumination and another reference image of uniform black illumination (a projector will always emit some amount of light). Then for each pixel in the camera image, the current value $g_i(u,v)$ is thresholded against the average of the white value $w(u,v)$ and the black value $b(u,v)$. In order to eliminate false classifications in areas that are

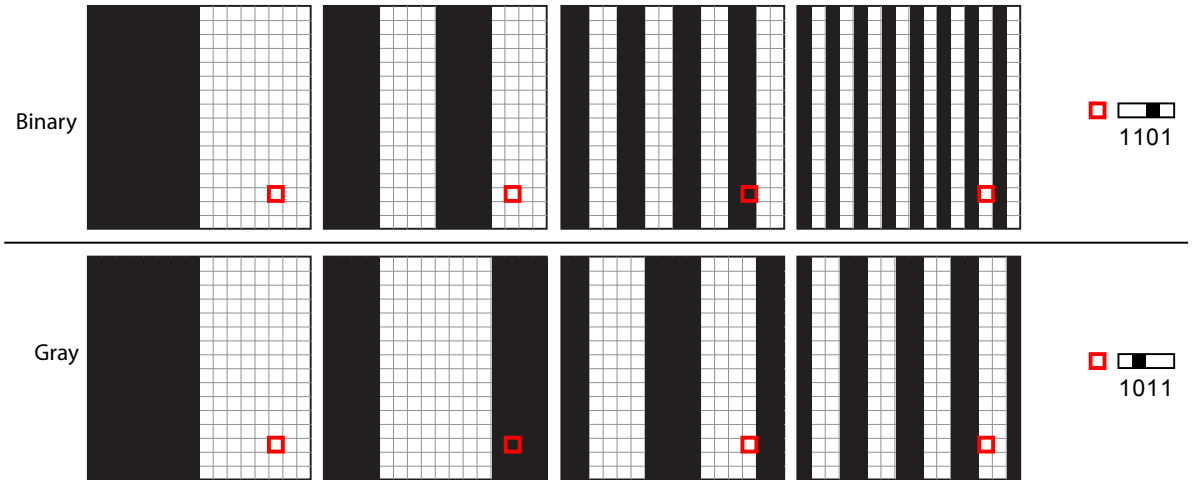

Figure 5.2: A vertical Gray code sequence encoding projector column indices. A temporal sequence of black and white corresponds to 0 or 1 bits in a Gray code sequence. The red square is at $[\text{row}, \text{col}] = [14, 14]$ and receives the binary sequence 1101 and the gray sequence 1011, both corresponding to $\text{col} = 14$.

not receiving any direct projection, the difference must be above some predefined threshold, λ.

$$b_i = g_i(u,v) < \frac{w(u,v)+b(u,v)}{2}, \quad \forall\{(u,v)|(w(u,v)-b(u,v)) > \lambda\} \tag{5.2}$$

Unfortunately, global illumination effects cause light to be scattered in non-trivial ways and this light is dependent on the projected pattern. In order to compensate for the scattered light, the pattern and its inverse is projected. Then the pixel is classified based on whether there was a positive or negative change between the pattern and the inverse. Thus for a $[1024, 768]$ resolution projector image there are 40 pattern images. Results from an example Gray code sequence (with inverse images) on a physical scene comprised of wooden blocks are shown in Figure 5.3. The revised classification scheme is,

$$b_i = g_i(u,v) < g_i^{-1}(u,v), \quad \forall(g_i(u,v) - g_i^{-1}(u,v)) > \lambda. \tag{5.3}$$

Gray codes are frequently used in digital signal processing and error correction and they have an added spatial benefit for projected patterns. Due to the limited dynamic range of a camera and the limited spatial precision of a projector, a white pixel will frequently bleed into a neighboring black pixel, causing blooming effects and an incorrectly decoded bit on borders between white and black lines. As seen in Figure 5.2 a binary pattern will result in borders occurring over and over again at the same pixel values, whereas in a Gray code pattern the borders are different in every image.

The codification technique discussed is inherently discrete and can only provide estimates for projector

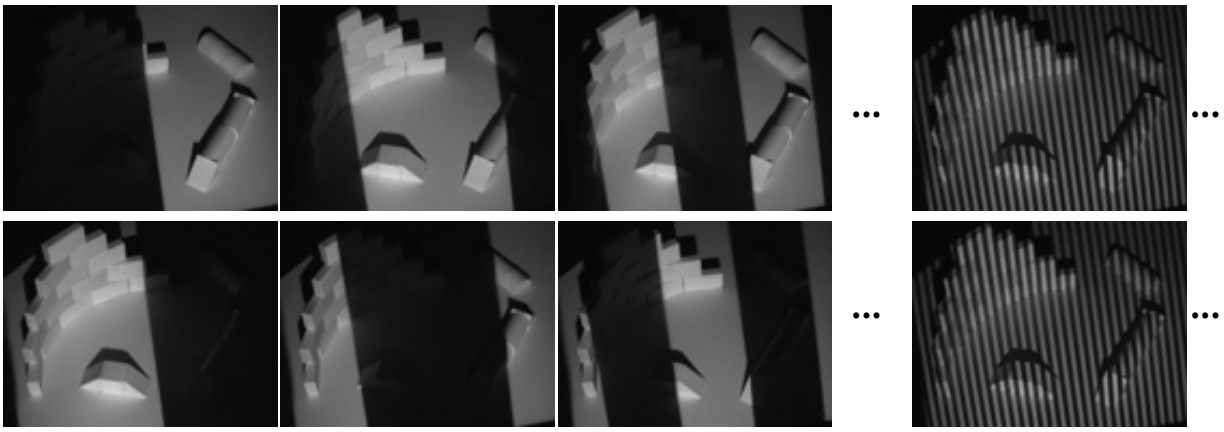

Figure 5.3: Results from Gray code projection onto a table sized set of white wooden blocks. (top row) Gray code pattern. (bottom row) Gray code pattern inverse.

pixel correspondences up to one pixel. If the resolution of the projector is greater than the camera, then the finest patterns cannot be decoded by the camera, see Figure 5.4. Also, due to sampling errors and the limited dynamic range of projectors and cameras the decoded signal is noisy on pattern borders.

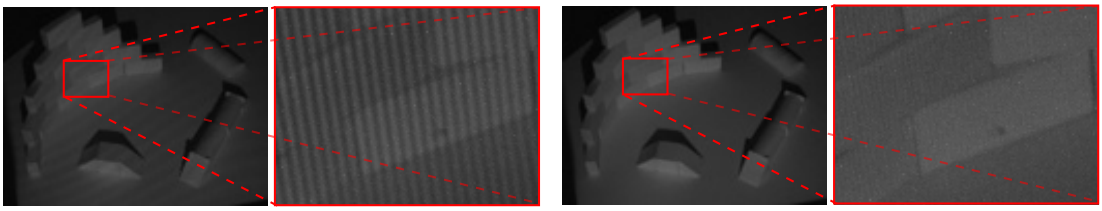

Figure 5.4: The final bits (9th and 10th) in a vertical Gray code sequence are extremely noisy as the camera resolution $[640,\ 480]$ is less than that of the projector $[1024,\ 768]$.

5.1.2 Sinusoidal Patterns

Gray code patterns are a binary signal and therefore do not utilize the full dynamic range of the projector. Through sending more information per pattern, the total number of patterns can be decreased and the accuracy of the resulting correspondences can be improved. Instead of sending a Gray code pattern, a horizontal and vertical phase shifted sinusoidal pattern is used to encode the projector row and column.

The three-step algorithm [61] utilizes three phase shifted sinusoidal patterns:

$$\begin{aligned} I_1(u,v) &= \mu(u,v) + A(u,v)\cos\left[\phi(u,v) - \alpha\right] \\ I_2(u,v) &= \mu(u,v) + A(u,v)\cos\left[\phi(u,v)\right] \\ I_3(u,v) &= \mu(u,v) + A(u,v)\cos\left[\phi(u,v) + \alpha\right] \end{aligned} \tag{5.4}$$

where $\mu(u,v)$ is the average intensity, $A(u,v)$ is the intensity modulation, $\phi(u,v)$ is the phase of the sinusoid and α is the phase step size. Typically $\alpha = \frac{\pi}{2}$ or $\alpha = \frac{2}{3}\pi$, in this work $\alpha = \frac{\pi}{2}$ is used. The three sinusoids are depicted in Figure 5.5.

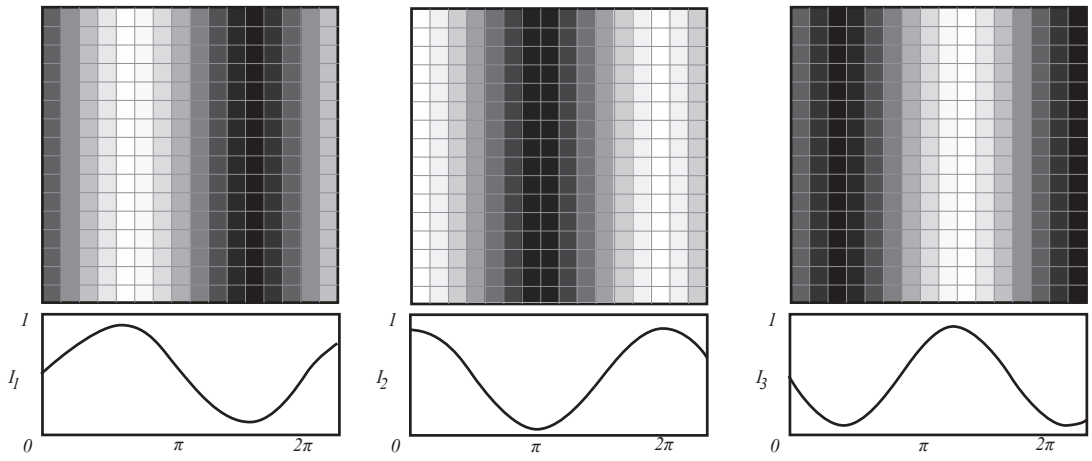

Figure 5.5: A vertical three step algorithm which utilizes sinusoidal patterns at three different phases $[-\frac{\pi}{2}, 0, \frac{\pi}{2}]$, where the wavelength is 14 pixels.

The system of three equations can be solved for the phase, up to a 2π ambiguity. The phase represents the pixel location, row or column, in the projector image. Typically a high frequency pattern is used because then global illumination effects are less noticeable across the camera image. A variety of wavelengths were tested and the pattern shown in Figure 5.5 uses $2\pi = 14$ pixels. The relative phase is,

$$\phi_r(u,v) = \arctan\left(\sqrt{3}\frac{I_1 - I_3}{2I_2 - I_1 - I_3}\right). \tag{5.5}$$

It is important to note that this solution was independent of surface albedo, A. One limitation of this method is that the above model assumes that the signal is linearly transmitted by the projector and linearly received by the camera. Unfortunately, projection and imaging devices typically suffer from non-linear warping which is usually modeled through a gamma curve [62]. In this work, gamma correction is omitted

limiting the precision of the sinusoidal patterns.

5.1.3 Hybrid Approach

We present a hybrid Gray code and sinusoidal pattern codification technique that can eliminate the 2π ambiguity in the sinusoidal patterns and can be used as a redundant error check. A Gray code pattern can be utilized to roughly locate the pixel up to the wavelength and the sinusoidal pattern can then determine the exact location. Thus the phase of the signal can be unwrapped from a relative phase to an absolute phase.

For instance, let us assume that the wavelength is 16 pixels and the projector resolution is $[1024,\ 768]$. If the column is being encoded, then the Gray code only needs to transmit $\log_2(1024/16) = 6$ bits of information, yielding 6 projected patterns. Then three sinusoidal patterns of wavelength 16 are projected, determining the relative phase. The Gray code and sinusoidal information are then combined to determine the absolute phase and the column is recovered. The absolute phase $\phi(u,v)$ is solved for using the relative phase $\phi_r(u,v)$ and the Gray code index $g(u,v)$.

$$\phi(u,v) = \phi_r(u,v,) + \lambda * g(u,v) \tag{5.6}$$

As the Gray code patterns have the most amount of noise on pattern edges, the sinusoidal wavelength and the Gray code resolution are chosen to have no common denominator. This technique can be further improved by ensuring that the sinusoidal results are close to the original Gray code estimates. The entire Gray code sequence is projected and both the sinusoidal and Gray code estimates are computed. Then if the distance between these two estimates is less than some threshold, α, it is deemed valid and the sinusoidal result is taken as the final value. Figure 5.6 shows a depiction of this process.

5.2 Triangulation

Acquiring the 3D geometry is fairly trivial with a calibrated projector-camera system and a set of dense correspondences. As shown in Figure 5.1, the ray through the camera pixel, x_c, and corresponding projector pixel, x_p, are triangulated and the intersection of these rays defines the 3D world point, X_w. With a calibrated camera and projector with projection matrices A^c and A^p respectively, along with

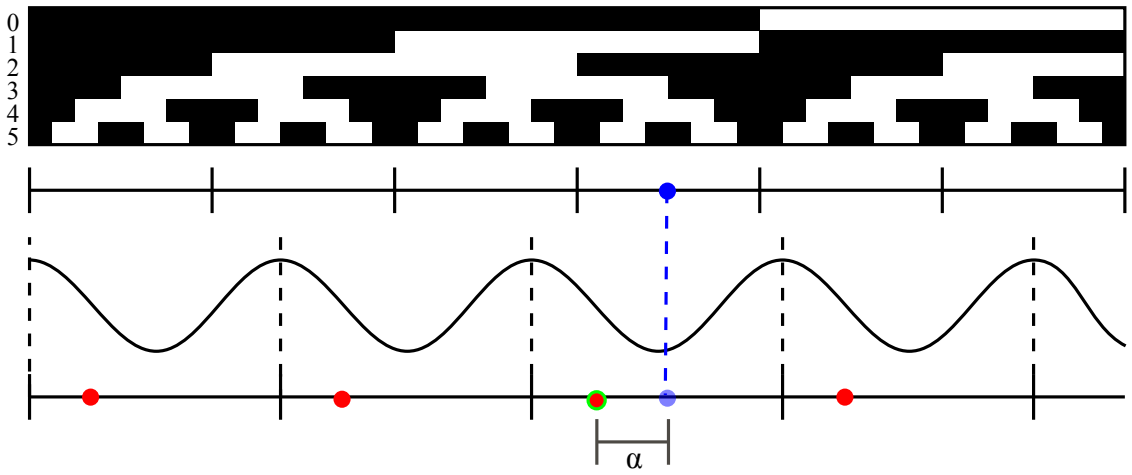

Figure 5.6: A Gray code sequence (top) first estimates the column value, shown in blue. Then a sinusoidal pattern (bottom) of wavelength 11 is used to determine the relative phase, shown in red. Then the absolute phase is determined using the Gray code pattern and the value is accepted if the Gray code and sinusoidal estimates are less than some threshold α.

correspondences $[u_i^c, v_i^c]$ and $[u_i^p, v_i^p]$, we have the following.

$$\begin{bmatrix} w^c u_i^c \\ w^c v_i^c \\ w^c \end{bmatrix} = \begin{bmatrix} a_{1,1}^c & a_{1,2}^c & a_{1,3}^c & a_{1,4}^c \\ a_{2,1}^c & a_{2,2}^c & a_{2,3}^c & a_{2,4}^c \\ a_{3,1}^c & a_{3,2}^c & a_{3,3}^c & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} w^p u_i^p \\ w^p v_i^p \\ w^p \end{bmatrix} = \begin{bmatrix} a_{1,1}^p & a_{1,2}^p & a_{1,3}^p & a_{1,4}^p \\ a_{2,1}^p & a_{2,2}^p & a_{2,3}^p & a_{2,4}^p \\ a_{3,1}^p & a_{3,2}^p & a_{3,3}^p & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{5.7}$$

Writing out the equations we have the following.

$$\left\{\begin{array}{rcl} w^c u_i^c & = & a_{1,1}^c X_i + a_{1,2}^c Y_i + a_{1,3}^c Z_i + a_{1,4}^c \\ w^c v_i^c & = & a_{2,1}^c X_i + a_{2,2}^c Y_i + a_{2,3}^c Z_i + a_{2,4}^c \\ w^c & = & a_{3,1}^c X_i + a_{3,2}^c Y_i + a_{3,3}^c Z_i + 1 \end{array}\right\}$$
$$\left\{\begin{array}{rcl} w^p u_i^p & = & a_{1,1}^p X_i + a_{1,2}^p Y_i + a_{1,3}^p Z_i + a_{1,4}^p \\ w^p v_i^p & = & a_{2,1}^p X_i + a_{2,2}^p Y_i + a_{2,3}^p Z_i + a_{2,4}^p \\ w^p & = & a_{3,1}^p X_i + a_{3,2}^p Y_i + a_{3,3}^p Z_i + 1 \end{array}\right\} \tag{5.8}$$

After substituting for w, we have the following.

$$\left\{\begin{array}{rcl} a_{3,4}^c u_i^c - a_{1,4}^c & = & (a_{1,1}^c - u_i^c a_{3,1}^c) X_i + (a_{1,2}^c - u_i^c a_{3,2}) Y_i + (a_{1,3}^c - u_i^c a_{3,3}) Z_i \\ a_{3,4}^c v_i^c - a_{2,4}^c & = & (a_{2,1}^c - v_i^c a_{3,1}^c) X_i + (a_{2,2}^c - v_i^c a_{3,2}) Y_i + (a_{2,3}^c - v_i^c a_{3,3}) Z_i \end{array}\right\}$$
$$\left\{\begin{array}{rcl} a_{3,4}^p u_i^p - a_{1,4}^p & = & (a_{1,1}^p - u_i^p a_{3,1}^p) X_i + (a_{1,2}^p - u_i^p a_{3,2}) Y_i + (a_{1,3}^p - u_i^p a_{3,3}) Z_i \\ a_{3,4}^p v_i^p - a_{2,4}^p & = & (a_{2,1}^p - v_i^p a_{3,1}^p) X_i + (a_{2,2}^p - v_i^p a_{3,2}) Y_i + (a_{2,3}^p - v_i^p a_{3,3}) Z_i \end{array}\right\} \tag{5.9}$$

We can then rewrite the equation in the form $PV = F$ where,

$$P = \begin{bmatrix} (a_{1,1}^c - u_i^c a_{3,1}^c) & (a_{1,2}^c - u_i^c a_{3,2}) & (a_{1,3}^c - u_i^c a_{3,3}) \\ (a_{2,1}^c - v_i^c a_{3,1}^c) & (a_{2,2}^c - v_i^c a_{3,2}) & (a_{2,3}^c - v_i^c a_{3,3}) \\ (a_{1,1}^p - u_i^p a_{3,1}^p) & (a_{1,2}^p - u_i^p a_{3,2}) & (a_{1,3}^p - u_i^p a_{3,3}) \\ (a_{2,1}^p - v_i^p a_{3,1}^p) & (a_{2,2}^p - v_i^p a_{3,2}) & (a_{2,3}^p - v_i^p a_{3,3}) \end{bmatrix}$$
$$V = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \text{ and } F = \begin{bmatrix} a_{3,4}^c u_i^c - a_{1,4}^c \\ a_{3,4}^c v_i^c - a_{2,4}^c \\ a_{3,4}^p u_i^p - a_{1,4}^p \\ a_{3,4}^p v_i^p - a_{2,4}^p \end{bmatrix} \tag{5.10}$$

We can then solve for V,

$$V = (P^t P)^{-1} P^t F. \tag{5.11}$$

Therefore, we can solve for the world space position, V, of the point given a calibrated projector-camera system.

5.3 Shape Modeling

Once the triangulation is complete, the result is a point cloud of world points stored as a camera depth map. As most applications utilize a mesh representation for objects, it would be useful to translate the point cloud of data into a set of triangles. Shape modeling from point clouds of complex surfaces is largely an open research area [63, 64, 65] and so simplification can make this problem more tractable.

As the point cloud was acquired from a single projector and camera in this application, a triangulation in camera space will be valid as long as the surface is continuous and there are no holes or errors in the scanning process. If the surface is discontinuous, then a triangulation in camera space may join triangles along object edges that are not physically realistic. For instance, if the scene was a coffee cup on a table, a camera based triangulation would join the side of the coffee cup to the adjacent table. To mitigate these effects, triangulation in camera space can be constrained to only connecting points with a 3D distance less than some threshold. Additionally, if there were holes in the scan there will be holes in the triangulation. While this triangulation procedure is far from ideal, it is computationally efficient and yields decent results for scans with a single projector and camera. If multiple projectors and cameras are used in order to acquire multiple views of the scene, then more complicated shape modeling approaches must be used.

The final scan contains high frequency noise at a very small scale, which leads to inconsistent triangle normals. In order to calculate surface normals the system applies principal component analysis to a rectangular neighborhood around each pixel in the depth map, smoothing the normals.

5.4 Results

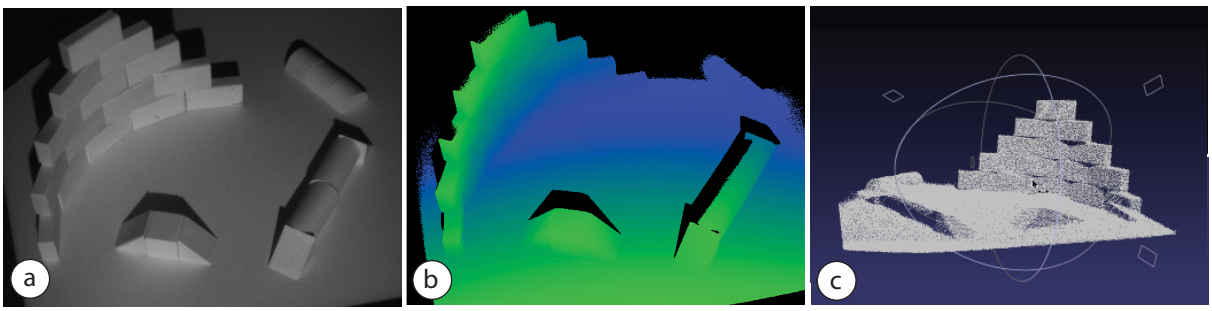

Figure 5.7: A high resolution scan of the surface [640, 480] is acquired with structured light and stored as a depth map. a) Original surface composed of wooden blocks of dimensions approximately [0.5m x 0.5m x 0.2m]. b) Depth map (near=green, far=blue). c) Acquired dense 3D point cloud (approximately 200,000 points).

There are a number of performance metrics that are important when evaluating a structured light scanning system [34]. This section describes the various metrics and the performance of our structured light system. These results were generated using the system described in 4.5.1. For context, an example scan with the system is shown in Figure 5.7. A set of wooden blocks of dimensions approximately [0.5m x 0.5m x 0.2m] was scanned yielding a dense point cloud of approximately 200,000 points.

5.4.1 Resolution

The resolution of the depth map is defined by the camera and projector resolution. With the current projector pattern codification technique, there is one projector-camera correspondence for every camera pixel. Therefore, the scan is currently limited by the camera resolution. For instance, a camera of resolution $[640, 480]$ would yield a maximum of $640 * 480 = 307,200$ points. The current system has been tested with cameras of resolution $[640, 480]$ and $[928, 616]$, and projectors of resolution $[1024, 768]$ and $[1200, 800]$.

As the pattern codification process is noisy due to the finite camera and projector dynamic range and resolution, there are holes in the resulting scan. A hole is defined as a camera pixel that does not decode a projector pixel, even though the projector and camera can both see a valid part of the surface. It is important to note that some camera pixels will correctly not decode a projector pixel. This can happen if the projector and camera cannot both see the pixel due to parallax, or if there is no surface within the scanning volume at this camera pixel. In a scan of a semi-rigid cardboard plane (see Figure 5.8), 80,061 pixels were valid, 79,900 were decoded with 161 holes. Therefore, 99.8 percent of the valid pixels were decoded.

Figure 5.8: Holes in the scan are generated due to noise in the pattern codification process. (a) The scene is comprised of a semi-rigid cardboard plane. (b) A hand drawn mask illustrating the portion of the camera image that should decode projector correspondences. White corresponds to a valid surface and black corresponds to invalid regions. (c) An image depicting the holes in the scan. White corresponds to decoded projector pixels, red corresponds to holes and black corresponds to invalid regions.

5.4.2 Acquisition Rate

The acquisition rate of the structured light scanner is limited by the camera frame rate, the projector frame rate, the synchronization of the projector and camera, the number of codification patterns and the processing time. Given the application requirements, the current system was developed for a high resolution scan with a low acquisition rate. While the projector frame rate is 60 Hz and the camera frame rate is 61 Hz, currently the projector and camera are not hardware synchronized. Therefore when a pattern is requested to be projected, the system must wait to capture a camera image until enough time has passed that the projector must be displaying the pattern. In the current implementation, the system stalls for 500 ms, giving the projector enough time to display the requested pattern. This delay could be almost entirely eliminated through custom hardware synchronization. With the hybrid Gray code and sinusoidal projector codification pattern, for a $[1024, 768]$ projector there are a total of 46 projector images. There are 20 images for the Gray code rows and 20 for the Gray code columns, which includes the inverse patterns. Then there are 3 images for the sinusoidal rows and 3 for the sinusoidal columns. With 46 images at 500 ms each, the total codification process takes 23 seconds. Finally, the un-optimized processing takes 2 seconds with an additional 5 seconds to write out the results in various formats.

5.4.3 Working Volume

Many structured light systems are evaluated at only one scale. The initial homography estimation in the projector-camera calibration step has enabled us to scan a wider variety of scenes. The system has been successfully used on objects as small as 0.3 meters in width and as large as 5 meters in width. Five representative example scans are shown in Figure 5.9. Scaling the system to larger objects, such as building exteriors is limited by the current calibration process.

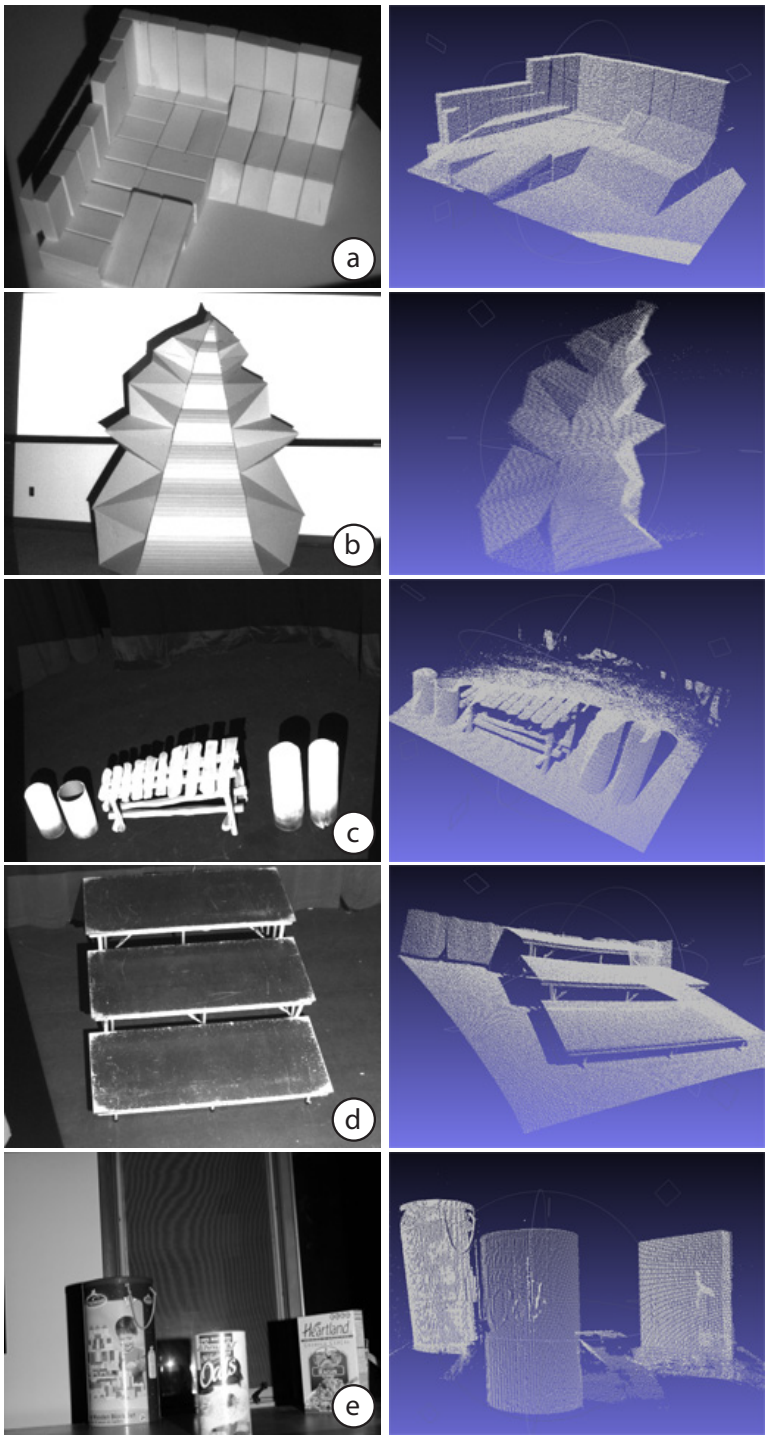

Figure 5.9: Five example structured light scans with dimensions [width, length, height]. a) Tabletop set of wooden blocks, approximately [0.5m x 0.5m x 0.2m]. b) A cardboard sculpture, approximately [2m x 3m x 1m]. c) Polystyrene sculptures, approximately [5m x 3m x 1m]. d) Wooden stage risers, approximately [3m x 4m x 1m]. e) Cardboard boxes and cylinders on a tabletop, approximately [1m x 1m x 0.5m].

5.4.4 Accuracy

The accuracy of a structured light system is influenced by systematic distortion and local noise. Systematic distortion (metric distortion) is caused by errors in the calibration process and local noise is typically caused by errors in the projector pattern codification process. Evaluating the errors in a structured light system requires some known geometry that can be used to compare the results.

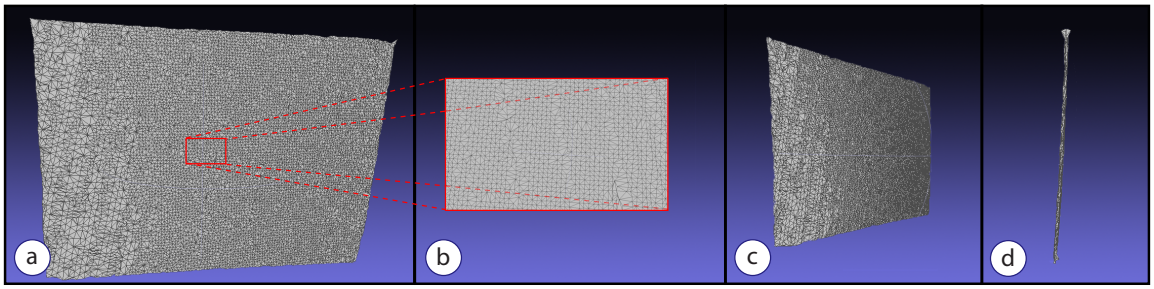

Figure 5.10: A semi rigid cardboard plane was scanned resulting in 239,700 points. (a) A triangulated data set which was subsampled for easier viewing. (b) A close up of a full resolution triangulation. (c) A perspective view of the plane. (d) A side view of the plane.

If the entire scene is composed of a single plane, then the resulting data should lie entirely in a single plane. Metric distortion would warp the plane, yielding a bent surface and local noise would slightly perturb the surface, yielding a rough plane. In order to evaluate the accuracy of the system, a plane was scanned yielding an example data set shown in Figure 5.10. Now we can fit a plane to this data set using RANSAC [66] and use the fit plane as an approximation of ground truth data. The deviation from this plane represents error in the structured light process. The root mean squared (RMS) error is commonly used as a representation for the overall error in a structured light system. The RMS error for our structured light system, was 1.5201 mm, see Figures 5.11 and 5.12. The projector-camera configuration was the same as shown in Figure 4.10. According to the computed extrinsic parameters, the projector was 1.71 m from the final calibration target and the camera was 1.52 m from the final calibration target. The plane had dimensions 0.91 m by 1.21 m. Therefore, an RMS error of 1.5201 mm corresponds to approximately a 0.1 percent error in the structured light scan based on the projector-camera distance to the scene.

With a 1.5201 mm RMS error on a scene of approximately $1m^3$, the accuracy of our structured light system is comparable with other research systems for low cost depth measurement. As the goal of our projector-camera system is ultimately surface augmentation, the results must be accurate enough for overlaying virtual content onto physical objects. The next chapter discusses the use of the acquired three dimensional geometry in surface augmentation and shows example surface augmentations.

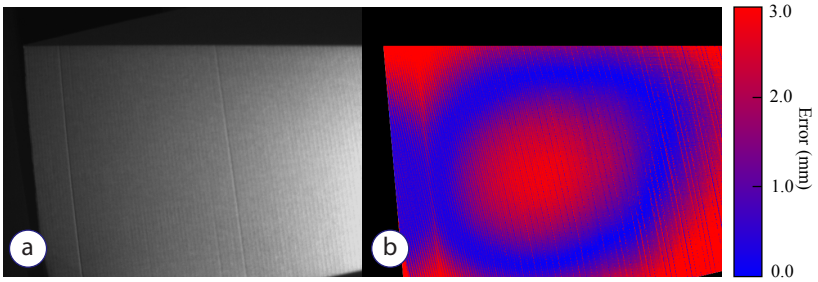

Figure 5.11: The root mean squared (RMS) error was 1.5201 mm in scanning a plane of dimensions [0.91 m x 1.21 m] at a distance of approximately 1.5 m. (a) A planar, semi-rigid cardboard sheet was scanned. (b) The distance to the ideal plane fit with RANSAC is depicted with blue corresponding to zero deviation and red corresponding to 3.0 mm of deviation.

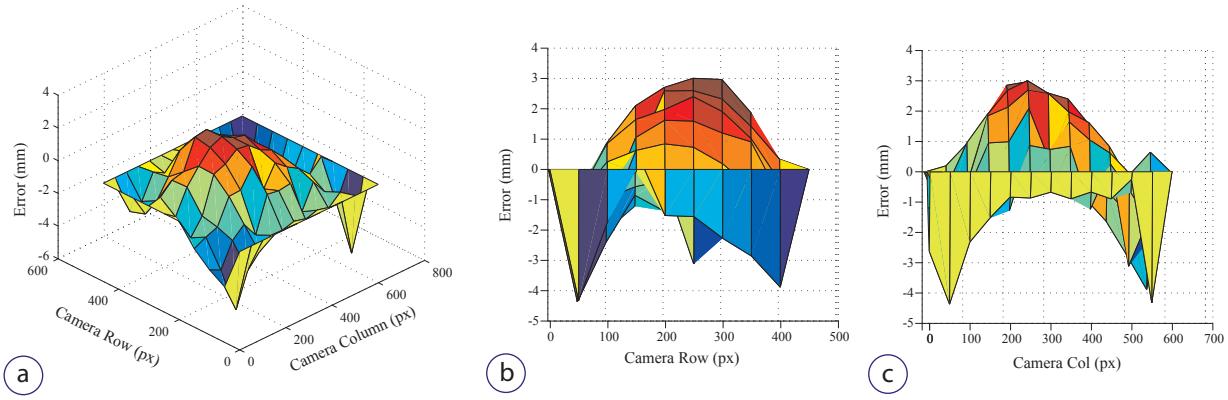

Figure 5.12: The signed distance from the measured plane to the ideal plane fit with RANSAC.

CHAPTER 6

SURFACE AUGMENTATION [1]

Augmenting surfaces through projection enables converting passive objects into display and interaction surfaces. Objects that were once passive can be made interactive, allowing for a host of new possibilities. In order to alter the surface of an object, the virtual model and the physical entity must be aligned. If the virtual geometry differs from the physical geometry, then there will be view dependent effects which prohibit simultaneous users and require complicated head tracking. Therefore, it is advantageous if the content remains constrained to the surface, eliminating view dependent effects. Additionally, it is unclear how surface augmentation content should be structured. Two dimensional content does not make use of the surface and three dimensional content is not constrained to a surface. In order to address these issues, we present interactive surface particles that are a view independent and surface independent content representation. In this chapter, the fundamental issues in surface augmentation are addressed and interactive surface particle sprites are presented.

6.1 Surface Augmentation Fundamentals

After a 3D scan is acquired, the original surface representation can be modified and textures can be replaced. In order to augment the surface, the virtual model is rendered from the point of view of the projector and then this image is displayed by the projector. If the projector is properly calibrated and the virtual model is accurate, then the virtual model will project directly onto the physical model. The virtual representation of the surface texture will be aligned with the physical surface texture. If the virtual surface texture is altered, then the physical texture will change, see Figure 6.1. Altering the projector image, changes the incident light on the physical model.

[1] Portions of this chapter were reprinted with permission. ©2010 IEEE. Reprinted, with permission, from the 8th IEEE International Symposium on Mixed and Augmented Reality, 2010, Build Your World and Play In It: Interacting with Surface Particles on Complex Objects, Jones, B.R., Sodhi, R., Campbell, R., Garnett, G., Bailey, B.P.

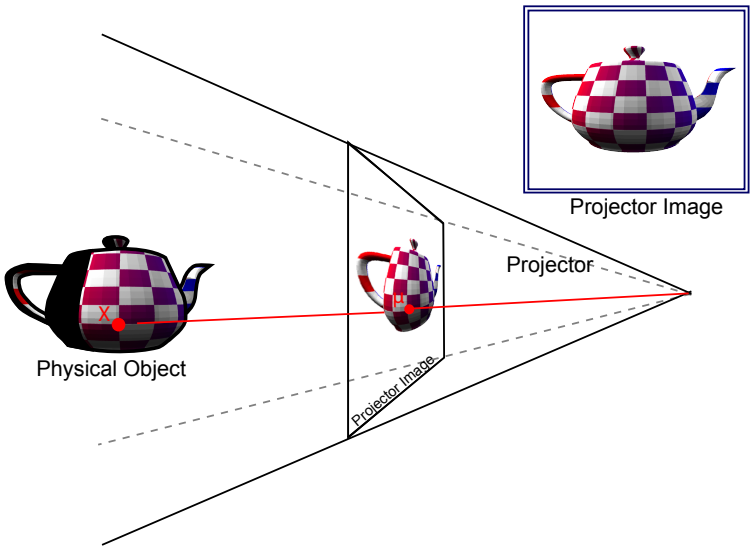

Figure 6.1: The texture on a physical teapot is augmented through projection. The virtual model is rendered from the projectors viewpoint, yielding the projector view in the upper right corner. This image is then displayed by the projector, which falls on the physical teapot. The world point on the physical object X is aligned with the corresponding image point μ.

6.1.1 Light Transport

In order to alter the surface texture, we must determine what color the projector should display given a specific desired surface appearance. The seminal Shader Lamps [6] paper offers a formal description of projector light transport which is condensed in the following description. Formally, in the rendering equation [67], the radiance at a point on a surface, X, that an observer in the direction (θ, ϕ) would see, can be modeled as:

$$L(X,\theta,\phi) = g(X,\theta,\phi)\,(L_e(X,\theta,\phi) + h(X,\theta,\phi)) \tag{6.1}$$

where

$$h(X,\theta,\phi) = \int_i F_r(X,\theta,\phi,\theta_i,\phi_i)\,L_i(X,\theta_i,\phi_i)\,cos(\theta_i)d\omega_i. \tag{6.2}$$

In this equation, $g(X,\theta,\phi)$ represents the geometry and accounts for visibility and distance, $L_e(X,\theta,\phi)$ represents the emitted radiance at the point, which is zero for non-light sources, and $F_r(X,\theta,\phi,\theta_i,\phi_i)$ is the Bidirectional Reflectance Distribution Function (BRDF) of the point. The BRDF of a point represents how light is absorbed and reflected over all incoming and outgoing directions. The $h(X,\theta,\phi)$ term is responsible for integrating over incident light angles, (θ_i,ϕ_i).

This model can be greatly simplified as $L_e(X,\theta,\phi) = 0$ (the surface does not emit light) and the

projector is assumed to act as a point light source a distance $d(X)$ away. Additionally, the surface is assumed to be diffuse, reflecting light equally in all directions, so the BRDF can be replaced with a constant diffuse coefficient $k_d(X)$. Given these assumptions the simplified equation is:

$$L'(X,\theta,\phi) = g(X,\theta,\phi)\; k_d(X)\; I_p(X,\theta_p,\phi_p)\; cos(\theta_p)/d(X)^2 \tag{6.3}$$

where $I_p(X,\theta_p,\phi_p)$ is the projector pixel in direction (θ_p,ϕ_p). The radiance $L'(x,\theta,\phi)$ can be created by projector radiance $I_p(X,\theta_p,\phi_p)$,

$$I_p(X,\theta_p,\phi_p) = \frac{L(X,\theta,\phi)\; d(X)^2}{k_d(X)\; cos(\theta_p)},\; \forall\{X|k_d(X) > 0\}. \tag{6.4}$$

Therefore, as long as the surface reflects some light at all the desired wavelengths, the radiance at a particular point can be controlled by changing the projector illumination correspondingly, also known as radiometric compensation [68, 69]. Unfortunately, projectors have a limited brightness and range, which limits the degree that surface texture can be augmented. In practice, a red saturated surface cannot be made to look bright green, as not enough green light is reflected from the surface and the projector brightness is limited.

This formulation can be arrived at more succinctly by starting with the Phong illumination model [70],

$$\begin{aligned} L(X) &= k_d(X)\; I_p(\mu)\; (n(X) \circ l_p) \\ L(X) &= k_d(X)\; I_p(\mu)\; cos(\theta_{n,l_p}) \end{aligned} \tag{6.5}$$

where I_p is the projector illumination at pixel μ in direction l_p, $n(X)$ is the surface normal at X and assuming n and l_p are unit length. The projector illumination required to create $L(X)$ is

$$I_p(\mu) = \frac{L(X)}{k_d(X)\; cos(\theta_{n,l_p})}. \tag{6.6}$$

6.1.2 Geometric Registration

Before surface augmentation is possible, the virtual model must be registered to the physical display surface. If the virtual model was obtained using structured light, then the projector is already calibrated with respect to the physical model. If an alternate 3D scanning method like a depth sensing camera was used, then the projector must be calibrated.

Projector calibration given a known model is essentially a 3D rig calibration as explained in Section 4.1. The easiest way to obtain correspondences is to have the user manually specify corresponding 3D world points and 2D projector correspondences. First, a set of uniquely identifiable feature points are identified in the virtual model and labeled. Then a mouse or other tracked input is used to select 2D projector points that correspond to the virtual feature points on the physical model. This technique was first outlined in Shader Lamps [6]. Following calibration, the virtual model can be rendered with the projector's intrinsic and extrinsic parameters.

The registration process is extremely sensitive as a single pixel difference will be visually apparent. With a single pixel shift, colors on the contours of objects that are intended for one object will appear on a different object. If the projector view is generated with the standard fixed function pipeline, then the intrinsic and extrinsic parameters define the projection and modelview matrices. If radial distortion is not modeled, then the projected image may deviate significantly from the physical object. This can be corrected through a non-linear warp implemented in a second render pass. First, the entire scene is rendered with the given intrinsic and extrinsic parameters. This rendered image is then used as a texture to a regular mesh that is warped to account for the radial distortion in the projector lens, see Figure 6.2.

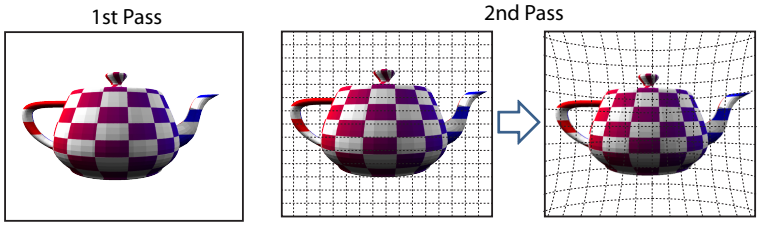

Figure 6.2: Radial distortion is corrected with a second render pass. The first pass incorporates the intrinsic and extrinsic parameters of the projector and the second pass warps the image to correct for the radial distortion.

6.1.3 View Dependent Effects

If the virtual geometry differs from the physical geometry, then the projection will appear to be warped on the display surface if viewed from another viewpoint. As shown in Figure 6.3, the projected image of the checkerboard, which is not part of the physical geometry, will be distorted from any other viewpoint.

It is possible to compensate for the user viewpoint through view dependent rendering, if the position of the user's head is known. Specifically, a set of virtual objects V can be rendered with other virtual objects that match the physical geometry G. First the scene is rendered with both sets of objects V, G from the

user's point of view. Then this image is used as a projective texture onto the geometry G which is rendered from the projectors view. Now, the physical view of the user should match the computed virtual view.

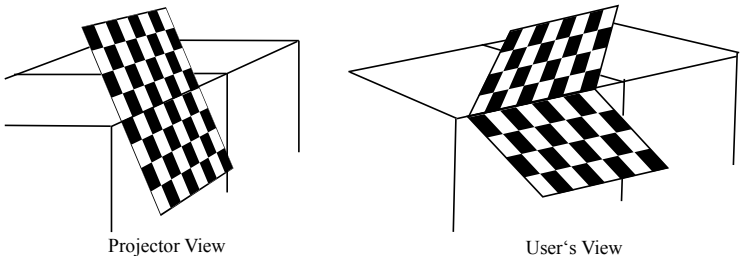

Figure 6.3: View dependent effects occur when the virtual and physical geometry differ. (left) A checkerboard quad is at a 45° angle with respect to two orthogonal planes. The view is rendered from the projectors point of view. (right) This view is displayed on the physical geometry through the projector and then viewed from an alternate angle. The resulting user view is distorted.

6.2 Surface Interaction Engine

Due to the limitations of view dependent effects, it is advantageous to author content that strictly adheres to the surface. Additionally, if Spatial Augmented Reality is to be scaled beyond single research demonstrations, content must be formulated in a reusable manner. In order to address these issues, we introduce interactive surface particles that enable view independent content that can be reused on any surface suitable for projection. Here we provide an overview of interactive surface particles and demonstrate their flexibility by showing three example applications on three sample surfaces. More information about the interactions enabled by the projector-camera system are covered in Sodhi's master's thesis [5].

Content is represented as surface constrained particles which are created programmatically without any knowledge of the final display surface. Our surface particle representation was motivated by research into using surface particles to visualize and control implicit surfaces [71, 72]. Along these lines, Su and Hart presented a programmable particle system, Wickbert, that incorporated a system of particles and behaviors upon which the Surface Interaction Engine (SIE) was modeled [73]. While Wickbert was built to achieve very different goals, the initial implementation of the SIE utilized the core Wickbert surface particle system.

Consider a scenario where an interaction designer adapts a miniature golf game for end users to play on a physical surface. The interaction designer programmatically creates interactive surface particles which include golf holes, balls, mats, sand pits and portals, along with associated interaction logic independent of

the final display surface. The end user then constructs a scene out of a set of wooden blocks, creating mounds and valleys, setting the stage for an upcoming golf tournament. The user then places portals, putting mats, balls and holes to create the scene. The user can then play the surface experience by directly interacting with the surface content using a stylus. For example, when the user putts a golf ball over a steep mound using a projected virtual putter, the ball slows, incorporating the physics of the physical surface. The whole process reveals a unique experience merging the virtual and physical worlds, enabling an immersive, tangible experience with interactions that are not possible in a truly physical environment.

6.3 Surface Particle Sprites

This scenario is made a reality through a surface particle sprite representation. A surface particle sprite has a position, velocity and acceleration on the surface and is subject to a set of forces and constraints. While the underlying system maintains 3D coordinates for the sprite, the SIE's Application Programming Interface (API) only allows the programmer to alter the sprite's 2D relative position on the surface. For instance, at any point on the surface, the particle can only move some degree of forward, backward, left and right. The relative coordinate system is based on the plane defined by the sprite, allowing the sprite to move only in the x_{local} and y_{local} directions, see Figure 6.4.

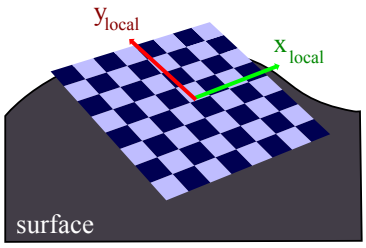

Figure 6.4: A surface particle is a 2D textured quad constrained to lie on a 3D surface and defined by a local 2D coordinate system (x_{local}, y_{local}).

Ideally, in order to determine the distance between sprites for interaction logic, the system would utilize geodesic distance (the shortest distance along a surface between two points). However, due to computational complexity, the system approximates geodesic distance with 3D Euclidean distance. Therefore the API allows querying the 3D position and orientation of a sprite, however only the relative 2D position, velocity and acceleration can be set by the programmer.

Interaction with surface particle sprites is constructed out of a series of entities, behaviors and events. Entities are created in an object-oriented manner with a class of sprites representing each type of element

in the scene. For instance, in a miniature golf game, golf balls would be represented as an entity class created by the programmer. Then the end user could place multiple golf ball sprite objects to create their unique golf course. The end user golf ball objects all share the same interaction logic as they are all objects of the same class. Behaviors define interactions with the surface that can apply to any sprite class and can therefore be reused. For instance, a drive behavior can take a joystick input and move a sprite on the surface. Also, every particle has a surface adhesion behavior attached which takes the current 3D position of the particle and attracts it to the surface at every frame. Events represent notifications of user input or important system state modification. For instance, a golf ball sprite needs to know if it has been clicked because someone is trying to putt the ball, or a golf ball sprite needs to know if it has collided with a hole because this indicates a successful putt. Sprites and behaviors can subscribe to notifications in the form of events.

6.4 The Build, Map, Play Process

After the interaction logic has been programmed by the designer, the surface content is ready for the end user. The user then constructs or finds a surface, maps content onto the surface and plays their experience, through a process entitled “Build, Map, Play”, see Figure 6.5.

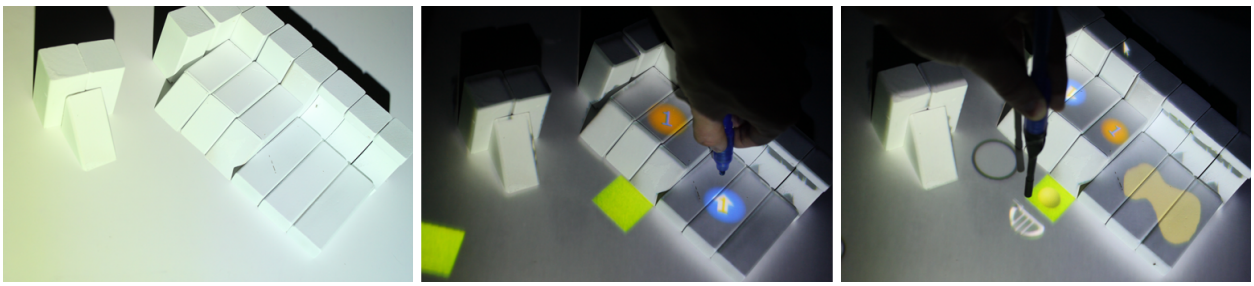

Figure 6.5: The “Build, Map, Play” process. Build (left): A user constructs a physical world out of wooden blocks. Map (center): The user places content, a miniature golf game, with a stylus. Play (right): The user interacts with the constructed physical surface, putting a golf ball across it.

6.4.1 Build

Part of the user experience of interacting with surface particles is constructing or finding a physical surface. The Surface Interaction Engine supports any physical surface that is opaque and light colored enough to receive projection, thus allowing for a wide range of interaction with the same interactive surface particles.

Example surfaces that have been used include cardboard, foam core, Styrofoam, sand, wooden blocks, plaster models and desks. While radiometric compensation techniques (Section 6.1.1) allow for projection onto colored surfaces, the Surface Interaction Engine leaves this for future work. Therefore, surfaces that are white and diffuse are ideal for receiving projected content from the system. After the surface is constructed or found, the system acquires a 3D model through the structured light scanning described previously. Other scanning methods could be employed including depth sensing cameras (time of flight), stereo vision or laser scanning. The system was implemented with structured light because a high resolution and relatively low noise scan is needed in order to accurately interact with a complex surface.

6.4.2 Map

Once the surface is constructed and scanned, the end user maps content to define a unique interactive experience on their surface. Mapping content is achieved using a stylus pen affixed with an IR LED. The SIE tracks the reflection of the IR light on the display surface and projects a virtual stylus cursor that corresponds to the 3D position and orientation of the cursor on the surface of the object. To map content, a radial menu appears on the surface surrounding the virtual cursor. The user can select sprite classes (entities) from the radial menu, which is composed of a gallery of scene elements representing entities that define the interactive experience. Clicking on the surface will place a new sprite which users can later scale, rotate or translate across the surface, thereby constructing the virtual world to overlay onto their physical world.

6.4.3 Play

Now the user can begin interacting with the surface particles, which react to the physics of the display surface. For example, the user may putt a virtual golf ball that rolls across some user placed wooden blocks incorporating the surface representation into the physics simulation. Then the golf ball may roll into a user placed portal and pop out at the top of another block construction. Depending on the content, a user may be able to map sprites as the interactive experience is running.

6.5 Results

In order to demonstrate that interactive surface particles allow for view independent and reusable content, we present three examples applications. As interactive surface particle content can be displayed on any

surface suitable for projection, the examples are demonstrated on three different surfaces, a user constructed model of wooden blocks, a sculptable sand pit and an ordinary office desk. While some examples are more practical on certain surfaces, we present each example on all three surfaces to prove generality. Each example has an accompanying image of the application on each example surface. As the images are capture from a different viewpoint than the projector, it is shown that the content is view independent.

6.5.1 Miniature Golf

The first example is a miniature golf game that is a mixture of physical and virtual game creation, see Figure 6.6. A virtual golf ball moves across the surface of the user constructed object, constrained to the surface and reacting to the physics of the object. Like traditional golf, the ball is affected by gravity and friction. However, in our version of miniature golf, momentum is preserved on the surface and the ball always adheres to the user constructed surface, allowing it to travel up walls. This unique physics model allows for a unique set of gameplay possibilities. When the user clicks on the virtual ball, a putter appears allowing the user to line up their stroke. The virtual world consists of golf balls, holes, putting mats, sand traps and portals. As an example, when played on wooden blocks, the user can quickly construct a challenging golf hole out of physical ramps, blocks and curves. After virtually placing a putting mat and a ball, the user can putt the ball up a wall, into a virtual portal which shoots the ball over another ramp and into the hole.

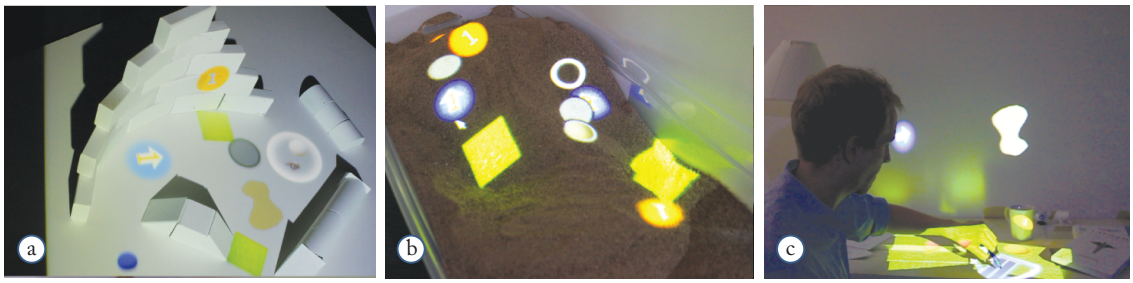

Figure 6.6: A miniature golf application where uses putt a virtual golf ball over physical surface. Presented on surfaces: a) wooden blocks b) a sandpit and c) a desk. The image is taken from a different viewpoint than the projector.

6.5.2 Photo Viewer

We created a simple photo viewer application which allows users to freely control rectangular image sprites on complex surfaces, see Figure 6.7. The photo viewer application provides a good demonstration of the types of interactions that can occur when traditional flat surface manipulations are simulated on complex physical surfaces, such as translation, rotation and scale. When used on an ordinary cubicle desk, users can utilize handy non-planar surfaces such as coffee cups, lamps, books, bobbleheads, etc. The user can place a recent picture of a visit to the local museum on a coffee cup and alternate pictures of a weekend family trip on a lamp shade.

Figure 6.7: A photo viewer application where images can be manipulated, flicked, attracted and repelled. Presented on surfaces: a) wooden blocks b) a sandpit and c) a desk. The image is taken from a different viewpoint than the projector.

6.5.3 Tank Game

A multi-player tank game explores collaboration and competition on a constructed physical surface, see Figure 6.8. Tanks move freely across the display object trying to destroy each other with bullets that wrap around surfaces. Tanks are driven by a joystick that rotates and translates the tank surface particle in its local 2D coordinate system. When played in a sand pit, a user could physically sculpt the mountains, valleys and rivers for an upcoming battle. A user may choose to sculpt a large valley around the mountain where they will place a virtual flag, providing a good defensive base. The user may plan on placing mines in the valley and the steep incline of their constructed mountain would slow the advance of enemy tanks.

As shown by the three example results, a surface particle representation allows for view independent, reusable surface augmentation. The images of the example applications demonstrate the view independence and the examples shown on each surface demonstrate the surface independence. Surface augmentation is only made possible by an accurate projector-camera calibration and structured light scan. The previous results (Sections 4.5.1, 5.4) demonstrated the accuracy of the underlying surface

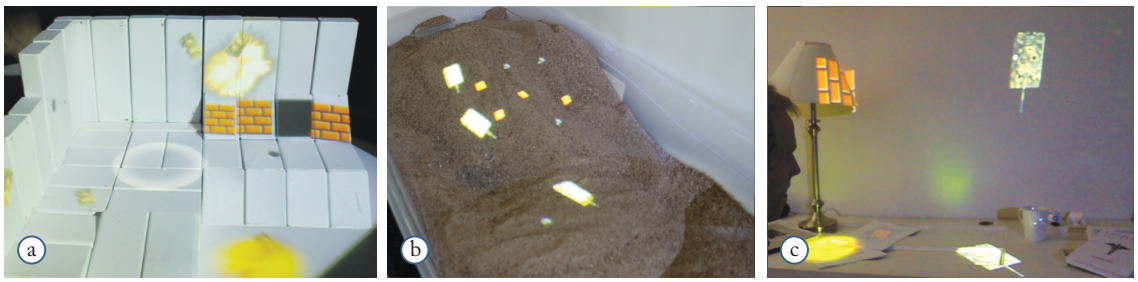

Figure 6.8: A multi-player tank battle application with bullets that wrap around corners. Presented on surfaces: a) wooden blocks b) a sandpit and c) a desk. The image is taken from a different viewpoint than the projector.

representation. More details about the interaction potential of a surface particle representation can be found in [5].

CHAPTER 7

CONCLUSION

We presented a complete, practical solution to surface augmentation with projector-camera systems. Our solution allows for calibration of real world scenes of varying scales, through an interactive calibration process with a novel initial homography estimate. We then described a unique hybrid gray code and sinusoidal pattern codification technique to acquire a dense set of projector-camera correspondences. The correspondences were then triangulated into a dense mesh representation of the surface. We presented results on the performance of the structured light system, along with a set of results in uncontrolled environments of varying sizes. Finally, interactive surface particles were introduced as a method for authoring content that is view independent and reusable on any surface suitable for projection.

There are still significant limitations to the current projector-camera system. The calibration process still requires a physical planar calibration target, which limits the system to room sized environments. With the rise of projection mapping onto building facades, it would be ideal if the scanning scaled to building sized environments. Current building projections rely on handcrafted models or original architectural models of the target building. Thus uncalibrated methods would have a large commercial impact. Additionally, radiometric compensation methods would enable projection onto a larger range of surfaces.

Real-time imperceptible structured light scanning would enable continuous surface re-evaluation, allowing users to manipulate and move the interaction surface. Alternatively, a real-time depth sensing camera such as the Microsoft Kinect[1] could be utilized for the scanning. If a real-time approach is adopted, there will be issues associating surface particles with a dynamically moving surface. For example, if a user decides to move a coffee cup that has been mapped with a photo, the real-time scanning must not only quickly acquire the geometry of the scene, but it must also utilize complex segmentation and classification algorithms to continuously relate content to objects.

Armed with a practical solution for surface augmentation on complex objects, the community is invited to consider the wealth of potential application areas. Point of sale displays can demonstrate the functionality of an object by overlaying information directly onto the sale item. Products can be

[1] Microsoft Kinect: http://www.xbox.com/en-US/kinect

demonstrated virtually, by augmenting the surface of the object as if the device was active. Household environments can become interactive spaces, opening computing applications in entirely new areas. Smart kitchens could display information and accept input on every available surface. For instance, a kitchen recipe could appear on a cabinet and upon request a user could be guided to an item in a cabinet through projected light. As demonstrated in the motivating examples, the Surface Interaction Engine unlocks an entirely new type of tangible gameplay experience. This could be extended to mobile computing scenarios utilizing the projectors that are now embedded in cell phones and laptops. Ad hoc games could be built on everyday objects and then played in a tangible virtual environment.

As the presentation covered fundamentals to implementation, this thesis should be an invaluable source for constructing a practical system for surface augmentation and depth scanning. As the system only utilizes low-cost, commodity components, surface augmentation is possible by anyone who can afford a projector. Hopefully, this thesis will help turn research efforts from building systems for accurate surface augmentation, towards considering the interaction potential of projector-camera systems and the host of potential applications.

REFERENCES

[1] O. Bimber, *Spatial augmented reality: Merging real and virtual worlds*. AK Peters Ltd, 2005.

[2] J. Rekimoto and M. Saitoh, "Augmented surfaces: a spatially continuous work space for hybrid computing environments," in *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. ACM, 1999, pp. 378–385.

[3] J. Underkoffler, B. Ullmer, and H. Ishii, "Emancipated pixels: real-world graphics in the luminous room," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 385–392.

[4] J. Ehnes, K. Hirota, and M. Hirose, "Projected Augmentation - Augmented Reality using Rotatable Video Projectors," *In Proc. Third IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 26–35, 2004.

[5] R. Sodhi, "Using Surface Particles to Interact with Complex Everyday Surfaces," Masters Thesis, University of Illinois at Urbana-Champaign, 2010.

[6] R. Raskar, G. Welch, K. Low, and D. Bandyopadhyay, "Shader lamps: Animating real objects with image-based illumination," in *Proceedings of the Eurographics Workshop in London, United Kingdom, June 25-27, 2001*. Springer Verlag Wien, 2001, p. 89.

[7] D. Bandyopadhyay, R. Raskar, and H. Fuchs, "Dynamic shader lamps: Painting on movable objects," in *Proceedings of the IEEE and ACM International Symposium on Augmented Reality*, 2001, pp. 207–216.

[8] C. Pinhanez, "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces," *Proceedings of Ubiquitous Computing*, 2001.

[9] R. Kjeldsen, A. Levas, and C. Pinhanez, "Dynamically reconfigurable vision-based user interfaces," *Mach. Vision Appl.*, vol. 16, no. 1, 2004.

[10] T. Johnson and H. Fuchs, "A Unified Multi-Surface, Multi-Resolution Workspace with Camera-Based Scanning and Projector-Based Illumination," in *IPT-EGVE Symposium*. The Eurographics Association, 2007.

[11] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image-based modeling and spatially immersive displays," in *Proceedings of the 25th annual conference on Computer Graphics and Interactive Techniques*. ACM, 1998, pp. 179–188.

[12] O. Bimber, A. Emmerling, and T. Klemmer, "Embedded entertainment with smart projectors," *Computer*, vol. 38, no. 1, pp. 48–55, 2005.

[13] H. Ishii, C. Ratti, B. Piper, Y. Wang, and A. Biderman, "Bringing clay and sand into digital design continuous tangible user interfaces," *BT Technology Journal*, vol. 22, no. 4, 2004.

[14] H. Ishii and B. Ullmer, "Tangible bits: Towards seamless interfaces between people, bits and atoms," *in Proc. CHI, SIGCHI Conf. Human Factors Comput. Syst., New York*, pp. 234–241, 1997.

[15] T. Johnson, H. Towles, A. State, and F. Wu, "A Projector-based Physical Sand Table for Tactical Planning and Review," *Training*, 2009.

[16] Y. Yasumuro, M. Imura, Y. Manabe, O. Oshiro, and K. Chihara, "Projection-based augmented reality with automated shape scanning," in *Proceedings of SPIE*, vol. 5664, 2005, p. 555.

[17] B. Caprile and V. Torre, "Using Vanishing Points for Camera Calibration," *The International Journal of Computer Vision*, vol. 4, no. 2, pp. 127–140.

[18] M. Pollefeys, R. Koch, and L. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," *International Journal of Computer Vision*, vol. 32, no. 1, pp. 7–25, 1999.

[19] R. Hartley, "Self-calibration from multiple views with a rotating camera," *Computer VisionECCV*, pp. 471–478, 1994.

[20] O. Faugeras and G. Toscani, "Camera calibration for 3D computer vision," in *International Workshop on Machine Vision and Machine Intelligence*, 1987, pp. 240–247.

[21] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. The MIT Press, 1993.

[22] O. Faugeras, Q. Luong, and S. Maybank, "Camera self-calibration: Theory and experiments," in *Computer Vision - ECCV*, no. 1. Springer, 1992, pp. 321–334.

[23] Z. Zhang, "A flexible new technique for camera calibration," *Proc. of IEEE Transactions oPattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2002.

[24] A. Majumder and R. Stevens, "Color nonuniformity in projection-based displays: analysis and solutions." *IEEE transactions on visualization and computer graphics*, vol. 10, no. 2, pp. 177–88, 2003.

[25] M. Brown, A. Majumder, and R. Yang, "Camera-based calibration techniques for seamless multiprojector displays." *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 193–206.

[26] R. Sukthankar, R. Stockton, and M. Mullin, "Smarter presentations: Exploiting homography in camera-projector systems," *In Proceedings of International Conference on Computer Vision*, 2001.

[27] H. Chen, R. Sukthankar, G. Wallace, and K. Li, "Scalable alignment of large-format multi-projector displays using camera homography trees," *Proc. of IEEE Vis*, 2002.

[28] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs, "Pixelflex2: A comprehensive, automatic, casually-aligned multi-projector display," in *IEEE International Workshop on Projector-Camera Systems*. Citeseer, 2003.

[29] R. Raskar, M. S. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs, "Multi-projector displays using camera-based registration," *In Proceeding of IEEE Visualization*, pp. 161–168, 1999.

[30] L. Wang, "Multi-Projector Display with Continuous Self-Calibration," *Computer*, vol. 1, no. 212, 2008.

[31] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines, "iLamps: geometrically aware and self-configuring projectors," in *ACM Transactions on Graphics*, vol. 223. ACM, 2003.

[32] J. Lee, S. Hudson, J. Summet, and P. Dietz, "Moveable Interactive Projected Displays Using Projector Based Tracking," in *UIST*, 2005, pp. 63–72.

[33] J. Lee, S. Hudson, and P. Dietz, "Hybrid infrared and visible light projection for location tracking," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007, p. 60.

[34] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno, "A low cost 3D scanner based on structured light," in *Computer Graphics Forum*, vol. 20, no. 3. John Wiley & Sons, 2001. [Online]. Available: http://www3.interscience.wiley.com/journal/118914580/abstract pp. 299–308.

[35] X. Armangué and J. Salvi, "Overall view regarding fundamental matrix estimation 1," *Image and vision computing*, vol. 21, no. 2, pp. 205–220, 2003.

[36] J. Posdamer and M. Altschuler, "Surface measurement by space-encoded projected beam systems," *Computer Graphics and Image Processing*, vol. 18, no. 1, pp. 1–17, Jan. 1982.

[37] M. Hashimoto and K. Sumi, "3-D Object Recognition Based on Integration of Range Image and Gray-scale Image," *System*, pp. 253–262.

[38] S. Zhang and P. S. Huang, "A novel method for structured light system calibration," *Optical Engineering*, vol. 45, 2006.

[39] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *CVPR*. IEEE, 2003, pp. 195–202.

[40] E. Horn and N. Kiryati, "Toward optimal structured light patterns," in *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*. IEEE, 2002, pp. 28–35.

[41] D. Caspi, N. Kiryati, and J. Shamir, "Range Imaging with Adaptive Color Structured Light," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 5, pp. 470–480, 2002.

[42] F. Zhou and G. Zhang, "Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations," *Image and Vision Computing*, vol. 23, pp. 59–67, 2005.

[43] C. Haihua, D. Ning, Y. Tianran, C. Xiaosheng, and L. Wenhe, "Calibration Algorithm for Structured Light 3D Vision Measuring System," pp. 324–328, 2008.

[44] J. Liao, "A Calibration Method for Uncoupling Projector and Camera of a Structured Light System," *Mechatronics*, pp. 770–774, 2008.

[45] R. Furukawa and H. Kawasaki, "Dense 3D Reconstruction with an Uncalibrated Stereo System using Coded Structured Light," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pp. 107–107, 2005.

[46] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM Transactions on Graphics*, vol. 25, no. 3, July 2006.

[47] D. Cotting, R. Ziegler, M. H. Gross, and H. Fuchs, "Adaptive instant displays: Continuously calibrated projections using per-pixel light control," *In Proceedings Eurographics*, pp. 705–714, 2005.

[48] K. Akasaka, R. Sagawa, and Y. Yagi, "A sensor for simultaneously capturing texture and shape by projecting structured infrared light," in *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, no. 3dim. IEEE, 2007, pp. 375–381.

[49] M. Vieira, L. Velho, A. Sa, and P. Carvalho, "A camera-projector system for real-time 3d video," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*. IEEE, 2005, p. 96.

[50] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. ACM, 2002, pp. 438–446.

[51] S. Zhang, "Recent progresses on real-time 3D shape measurement using digital fringe projection techniques," *Optics and Lasers in Engineering*, vol. 48, no. 2, pp. 149–158, Feb. 2010.

[52] S. Narasimhan, S. Koppal, and S. Yamazaki, "Temporal dithering of illumination for fast active vision," *Computer VisionECCV 2008*, pp. 830–844, 2008.

[53] D. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[54] R. Szeliski, *Computer Vision : Algorithms and Applications*. Springer, 2010.

[55] J. Batlle and E. Mouaddib, "Recent progress in coded structured light as a technique to solve the correspondence problem: a survey," *Pattern recognition*, vol. 31, no. 7, pp. 963–982, 1998.

[56] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[57] S. Jordan and M. Greenspan, "Projector Optical Distortion Calibration Using Gray Code Patterns," *Computer Engineering*, 2010.

[58] S. Audet and M. Okutomi, "A user-friendly method to geometrically calibrate projector-camera systems," *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 47–54, June 2009.

[59] D. Lanman and G. Taubin, "Build your own 3D scanner: optical triangulation for beginners," in *SIGGRAPH ASIA 2009 Courses*, 2009, pp. 1–94.

[60] J. Salvi and J. Batlle, "Pattern codification strategies in structured light systems," *Techniques*.

[61] P. S. Huang and S. Zhang, "Fast three-step phase-shifting algorithm." *Applied optics*, vol. 45, no. 21, pp. 5086–91, July 2006.

[62] A. Majumder, "Properties of Color Variation Across a Multi-Projector Display," *Proceedings of SID Eurodisplay*, vol. 2002, p. 3, 2002.

[63] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans, "Reconstruction and representation of 3D objects with radial basis functions," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, no. August. ACM, 2001, pp. 67–76.

[64] S. Fleishman, "Robust moving least-squares fitting with sharp features," *ACM Transactions on Graphics*, vol. 24, no. 3, p. 544, July 2005.

[65] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*. Eurographics Association, 2006, pp. 61–70.

[66] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[67] J. Kajiya, "The rendering equation," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 143–150.

[68] A. Grundhöfer and O. Bimber, *Real-time adaptive radiometric compensation.*, 2008, vol. 14, no. 1.

[69] O. Bimber, F. Coriand, A. Kleppe, E. Bruns, S. Zollmann, and T. Langlotz, "Superimposing pictorial artwork with projected imagery," *IEEE Multimedia*, vol. 12, no. 1, pp. 16–26, 2005.

[70] B. Phong, "Illumination for computer generated pictures," *Communications of the ACM*, vol. 18, no. 6, pp. 311—-317, 1975.

[71] A. Witkin and P. Heckbert, "Using particles to sample and control implicit surfaces," *SIGGRAPH*, pp. 269–277, 1994.

[72] J. Hart, E. Bachta, W. Jarosz, and T. Fleury, "Using particles to sample and control more complex implicitsurfaces," *Shape Modeling International, 2002.*, 2002.

[73] W. Su and J. Hart, "A programmable particle system framework for shape modeling," *International Conference on Shape Modeling and Applications (SMI')*, pp. 114–123, 2005.